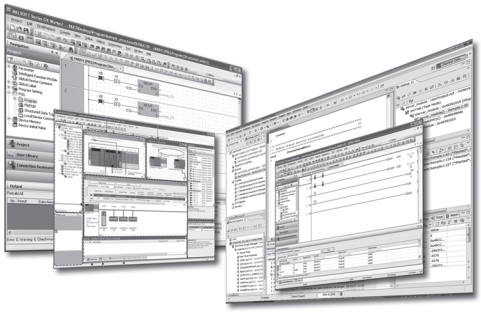
MITSUBISHI ELECTRIC

MELSOFT

Engineering Software

GX Works3 Operating Manual

-SW1DND-GXW3-E



SAFETY PRECAUTIONS

(Read these precautions before using this product.)

Before using this product, please read this manual carefully and pay full attention to safety to handle the product correctly.

The precautions given in this manual are concerned with this product only. For the safety precautions for the programmable controller system, refer to the user's manual for the module used and the MELSEC iQ-R Module Configuration Manual.

In this manual, the safety precautions are classified into two levels: "⚠WARNING" and "⚠CAUTION".

⚠ **WARNING**	Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.
⚠ **CAUTION**	Indicates that incorrect handling may cause hazardous conditions, resulting in minor or moderate injury or property damage.

Under some circumstances, failure to observe the precautions given under "⚠CAUTION" may lead to serious consequences.

Observe the precautions of both levels because they are important for personal and system safety.

Make sure that the end users read this manual and then keep the manual in a safe place for future reference.

[Design Precautions]

⚠ WARNING

- When data change, program change, or status control are performed from a personal computer to a running CPU module, create an interlock circuit outside the programmable controller to ensure that the whole system always operates safely.
  Furthermore, for the online operations performed from a personal computer to a programmable controller CPU, the corrective actions against a communication error due to such as a cable connection fault should be predetermined as a system.

[Startup and Maintenance Precautions]

⚠ CAUTION

- The online operations performed from a personal computer to a running CPU module (program change while a CPU module is in RUN, operating status changes such as RUN-STOP switching, and remote control operation) must be executed after the manual has been carefully read and the safety has been ensured.
- When changing a program while a CPU module is in RUN (online program change), it may cause a program corruption in some operating conditions. Fully understand the precautions before use.
- The positioning test functions such as OPR, JOG, inching, or testing positioning data for positioning module must be executed with the CPU module set to STOP after the manual has been carefully read and the safety has been ensured.
  Specially when executing the function on the network system, ensure the safety thoroughly since the machinery whose operation cannot be checked by an operator may be activated. The operation failure may cause the injury or machine damage.

CONDITIONS OF USE FOR THE PRODUCT

(1) Mitsubishi programmable controller ("the PRODUCT") shall be used in conditions;
i) where any problem, fault or failure occurring in the PRODUCT, if any, shall not lead to any major or serious accident; and
ii) where the backup and fail-safe function are systematically or automatically provided outside of the PRODUCT for the case of any problem, fault or failure occurring in the PRODUCT.

(2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries.
MITSUBISHI SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI'S USER, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.
("Prohibited Application")
Prohibited Applications include, but not limited to, the use of the PRODUCT in;

- Nuclear Power Plants and any other power plants operated by Power companies, and/or any other cases in which the public could be affected if any problem or fault occurs in the PRODUCT.
- Railway companies or Public service purposes, and/or any other cases in which establishment of a special quality assurance system is required by the Purchaser or End User.
- Aircraft or Aerospace, Medical applications, Train equipment, transport equipment such as Elevator and Escalator, Incineration and Fuel devices, Vehicles, Manned transportation, Equipment for Recreation and Amusement, and Safety devices, handling of Nuclear or Hazardous Materials or Chemicals, Mining and Drilling, and/or other applications where there is a significant risk of injury to the public or property.

Notwithstanding the above, restrictions Mitsubishi may in its sole discretion, authorize use of the PRODUCT in one or more of the Prohibited Applications, provided that the usage of the PRODUCT is limited only for the specific applications agreed to by Mitsubishi and provided further that no special quality assurance or fail-safe, redundant or other safety features which exceed the general specifications of the PRODUCTs are required. For details, please contact the Mitsubishi representative in your region.

INTRODUCTION

Thank you for purchasing the Mitsubishi FA software, MELSOFT series.

This manual describes the programming and functions required when using GX Works3.

Before using this product, please read this manual carefully, and develop familiarity with the functions and performance of GX Works3 to handle the product correctly.

CONTENTS

SAFETY PRECAUTIONS 1
CONDITIONS OF USE FOR THE PRODUCT 2
INTRODUCTION 3
RELEVANT MANUALS 12
TERMS 12

PART 1 FUNDAMENTALS OF GX Works3

CHAPTER 1 BEFORE USING THIS PRODUCT 16

1.1 Main Functions of GX Works3 16
1.2 RnSFCPU Supported Functions 19
Display of safety data identification 20
1.3 Procedure from Project Creation to CPU Module Operation 21
1.4 Learning Operation Method of GX Works3 24
Displaying Help 24
Connecting to MITSUBISHI ELECTRIC FA Global Website 24
Checking version of GX Works3 24
1.5 Connection Configurations between Personal Computer and CPU Module 25

CHAPTER 2 SCREEN CONFIGURATION AND BASIC OPERATIONS 27

2.1 Starting and Exiting GX Works3 27
2.2 Switching Displayed Language 27
2.3 Screen Configuration 28
Main frame 28
Navigation window 31
Element Selection window 32
2.4 Menu List 35
2.5 Checking and Changing Shortcut Keys 47
2.6 Checking and Changing Colors and Fonts 48
2.7 Setting Comment Display 49
2.8 Setting Options 50
2.9 Printing Data 51

PART 2 SYSTEM DESIGN AND SETTINGS

CHAPTER 3 PROJECT MANAGEMENT 56

3.1 Project File and Data Configuration 56
File format 56
Backup of safety project 57
Data configuration 58
3.2 Creating Project File 61
Creating project 61
Opening project 63
Opening GX Works2 format project 64
Saving project 69
Deleting project 70

3.3 Creating Data . . . 71
Creating projects . . . 72
Editing projects . . . 73
Property . . . 74
3.4 Changing Module Type of Project . . . 75
3.5 Verifying Projects . . . 77
Performing verification . . . 77
Checking verification result . . . 78
Exporting data to file . . . 79
3.6 Project Revision . . . 80
Registering revision information . . . 80
Displaying revision list . . . 81
3.7 Managing Profile . . . 83

CHAPTER 4 CREATING MODULE CONFIGURATION DIAGRAM AND SETTING PARAMETERS 84

4.1 Creating Module Configuration Diagram . . . 85
Editor configuration of Module Configuration Diagram . . . 85
Placing objects . . . 86
Checking module information . . . 88
Setting parameters on the Module Configuration Diagram . . . 89
Multiple CPU configuration . . . 90
Inputting start XY in batch . . . 92
Inputting default points in batch . . . 92
Checking power supply capacity and I/O points . . . 92
Checking system configurations . . . 92
Operation for Write to PLC/Read from PLC . . . 92
4.2 Setting Parameters . . . 93
Common operations of parameter settings . . . 94
Setting system parameters . . . 95
Setting parameters of control CPU . . . 96
Setting parameters of I/O module and Intelligent function module . . . 97
Parameter interaction with MELSOFT Navigator . . . 101
4.3 Other settings of Intelligent function module . . . 102
Setting by module tool . . . 102
Simple motion module setting tool . . . 103
4.4 Predefined Protocol Support Function . . . 104
Starting and exiting predefined protocol support function . . . 104
4.5 Circuit Trace Function . . . 105

CHAPTER 5 REGISTERING LABELS 106

5.1 About Labels . . . 106
5.2 Registering Labels . . . 107
Configuration of label editor . . . 107
Entering information . . . 109
Creating structure definition . . . 114
Importing/exporting files . . . 114
5.3 Registering Module Labels . . . 115
5.4 Registering System Label . . . 116
System label . . . 116

Registering labels in system label database . . . 117
Importing system labels in system label database to GX Works3 . . . 117
Releasing system labels . . . 117
Verifying system label information . . . 118
Importing the changes of system label database . . . 118

PART 3 PROGRAMMING

CHAPTER 6 CREATING PROGRAMS 120

6.1 Programming Functions . . . 120
Creation procedure . . . 121
6.2 Program execution order/execution type settings . . . 121
Program execution order settings . . . 121
Program execution types . . . 122
6.3 Creating Ladder Programs . . . 123
Configuration of ladder editor . . . 123
Entering ladder programs . . . 127
Entering comments/statements/notes . . . 135
Inserting/deleting NOP instruction . . . 139
Copying/pasting elements . . . 139
Changing ladder blocks temporarily . . . 141
Searching for/replacing data in program . . . 143
Displaying instruction help . . . 144
6.4 Creating ST Programs . . . 145
Configuration of ST editor . . . 145
Entering programs . . . 147
Searching for/replacing data in program . . . 150
Displaying instruction help . . . 150
6.5 Creating FBD/LD Programs . . . 151
Configuration of FBD/LD editor . . . 151
Entering programs . . . 154
Searching for/replacing data in programs . . . 157
Displaying help . . . 157
6.6 Creating SFC Programs . . . 158
Configuration of SFC diagram editor . . . 159
Creating SFC diagrams . . . 163
Creating/displaying Zooms (action/transition) . . . 172
Displaying an SFC block list . . . 173
Searching for/replacing programs . . . 175
Displaying the Help . . . 175
6.7 Registering Device Comments . . . 176
Device comments . . . 176
Configuration of device comment editor . . . 178
Creating device comments . . . 179
Deleting unused device comments . . . 180
Clearing all device comments . . . 180
Importing to/exporting from files . . . 181
Searching device comments . . . 181
Reading sample comments . . . 182
6.8 Checking Programs . . . 183

6.9 **Converting Programs** 184
Converting/converting all programs 184
Performing online program change simultaneously with conversion 189
Checking for errors and warnings 189
6.10 **Calculating Memory Size** 190

CHAPTER 7 SETTING DEVICE MEMORY 191

7.1 **About Device Memory** 191
7.2 **Configuration of Device Memory Editor** 192
Setting display format 192
7.3 **Setting Device Memory** 193
Setting device values in units of points 193
Setting values in batch 194
Setting character strings 195
Clearing whole memory of device memory 195
Interaction with device initial value 196
7.4 **Writing/Reading Data to/from CPU Module** 196

CHAPTER 8 SETTING DEVICE INITIAL VALUES 197

8.1 **About Device Initial Values** 197
8.2 **Configuration of Device Initial Value Editor** 197
8.3 **Setting Device Initial Values** 198

CHAPTER 9 SEARCHING DATA 200

9.1 **Searching/Replacing Devices/Labels** 202
9.2 **Searching/Replacing Instructions** 204
9.3 **Searching/Replacing Character Strings** 205
9.4 **Changing Contacts Between Open Contact and Close Contact** 207
9.5 **Batch Replacing of Devices and Labels** 208
9.6 **Displaying Reference Information List of Devices/Labels** 209
Creating/displaying cross reference information 210
9.7 **Displaying Device Usage List** 212

CHAPTER 10 SEGMENTING PROGRAMS 213

10.1 **Creating Function Blocks** 214
Function blocks 214
Creating function blocks 214
Enhanced use of module FBs 216
10.2 **Creating Functions** 217
Functions 217
Creating functions 217
10.3 **Enhancing Use of User Library** 219
User library 219
Creating libraries 220
Enhanced use of libraries 221
10.4 **Enhanced Use of Application Library/Sample Library** 223

PART 4 DEBUGGING AND OPERATION

CHAPTER 11 SIMULATING PROGRAM 226

11.1 Simulation Function . . . 226
Safety and handling consideration . . . 226
11.2 Starting/Stopping Simulation . . . 227
Cooperative simulation . . . 227
11.3 External Device Operation Simulation . . . 229
I/O System Setting function . . . 229
Execution procedure of I/O System Setting function . . . 229
Executing I/O System Setting function . . . 229
Supported device/label . . . 233

CHAPTER 12 SETTING ROUTE TO CPU MODULE 235

12.1 Specification of Connection Destination . . . 235
12.2 Direct Connection . . . 237
USB connection . . . 237
Ethernet connection . . . 237
Serial connection . . . 239
12.3 Accessing via Network (Single Network) . . . 240
12.4 Accessing via Network (Co-existence Network) . . . 241
12.5 Accessing via Serial Communication Module . . . 242
Connection on a 1:1 basis . . . 242
Connection on a 1:n basis . . . 243
12.6 Accessing via GOT (GOT Transparent Function) . . . 244
Accessing via GOT . . . 244
Accessing via module . . . 244
12.7 Accessing to Multiple CPU System . . . 245
12.8 Considerations of Communication with CPU module . . . 247

CHAPTER 13 WRITING/READING DATA TO CPU MODULE 248

13.1 Writing/Reading Programmable Controller Data . . . 248
Configuration of Online Data Operation screen . . . 248
Writing data to programmable controller . . . 250
Reading data from programmable controller . . . 254
Deleting data in CPU module . . . 256
Considerations for Online data operation . . . 256
13.2 Verifying Programmable Controller Data . . . 257
13.3 Writing Programs While CPU is in RUN . . . 258
Considerations . . . 258
Writing data after changing programs partially while CPU module is in RUN (Online program change) . . . 259
Writing data in units of files while the CPU module is in RUN (file batch online change) . . . 261
13.4 Reading/Writing/Deleting User Data . . . 262
13.5 Deleting All Files (Initialization) . . . 263
13.6 Writing/Reading Data to/from Memory Card . . . 264
Configuration of Memory Card Operation screen . . . 264
Writing to/Reading from memory card . . . 264

CHAPTER 14 CHECKING OPERATION OF PROGRAMS 265
14.1 Monitor Status . . . 266
14.2 Check Execution Programs on Program Editor . . . 268
Ladder . . . 269
ST . . . 270
FBD/LD . . . 271
SFC . . . 272
14.3 Checking Device/Buffer Memory in Batch . . . 275
14.4 Checking Current Values by Registering Devices/Labels . . . 277
14.5 Checking Processing Time of Program . . . 280
14.6 Checking Execution Counts of Interrupt Programs . . . 281
14.7 Checking Current Values in Intelligent Function Module . . . 282
14.8 Saving and Checking Device Value . . . 284
How to use the Memory Dump function . . . 284
Setting internal buffer capacity for the Memory Dump function . . . 284
Writing memory dump settings . . . 285
Reading memory dump results . . . 286
Displaying memory dump results . . . 287
14.9 Checking Sampled Data on Program Editor . . . 288

PART 5 MAINTENANCE AND INSPECTION

CHAPTER 15 PROTECTING DATA 292
15.1 Preventing Illegal Access to Program (Protected by Password) . . . 293
Block password function . . . 293
How to use block password . . . 293
Setting block password . . . 293
15.2 Preventing Illegal Access to Program (Protected by Key) . . . 295
Security key authentication function . . . 295
How to use the Security key authentication function (access prevention) . . . 296
Creating/deleting security key . . . 297
Copying security key . . . 298
Registering security key in program file . . . 299
15.3 Preventing Illegal Program Execution . . . 300
How to use the Security key authentication function (execution prevention) . . . 300
Writing/deleting security key to/from CPU module . . . 301
15.4 Preventing Illegal Access to Project . . . 303
User Authentication function . . . 303
How to use the User Authentication function (access prevention) . . . 304
Logging on to project . . . 304
Changing password of logon user . . . 304
User management . . . 305
15.5 Preventing Illegal Access to CPU Module . . . 306
How to use the User Authentication function (access prevention) . . . 306
Writing/reading user information to/from CPU module . . . 306
Logging on to CPU module . . . 306
Changing password . . . 306
15.6 Preventing Illegal Data Reading/Writing . . . 307
File password function . . . 307

How to use file password . . . 308
Setting file password . . . 309
15.7 Restricting Access from Other Than Specific Communication Route . . . 310
Remote password function . . . 310
How to use remote password . . . 310
Setting remote password . . . 311

CHAPTER 16 DIAGNOSING MODULE 312

16.1 Checking Module Status of System . . . 313
16.2 Diagnosing CPU Module . . . 314
16.3 Diagnosing Network . . . 316
Ethernet diagnostics . . . 316
CC-Link IE Controller Network diagnostics . . . 317
CC-Link IE Field Network diagnostics . . . 319
MELSECNET diagnostics . . . 320
CC-Link diagnostics . . . 321
16.4 Checking Error History/Operation History . . . 322

CHAPTER 17 SAMPLING DEVICE DATA 323

17.1 Data Logging Function . . . 323
How to use the data logging function . . . 323
Setting internal buffer capacity for the data logging function . . . 323

CHAPTER 18 CHECKING/CHANGING CPU MODULE OPERATION 325

18.1 Setting Clock in CPU Module . . . 325
18.2 CPU Module Remote Operation . . . 326
18.3 Operations of RnSFCPU . . . 327
Switch Safety Operation Mode . . . 327
Check Safety Data Identity . . . 329
18.4 Checking Memory Usage . . . 330
18.5 Initializing/Clearing Memory . . . 331

APPENDIX 332

Appendix 1 Functions Added to/Changed from Previous Version . . . 332
Appendix 2 Unusable Character String (Reserved Word) . . . 334
Appendix 3 Applicable Devices in GX Works3 . . . 337
For RCPU . . . 337
For FX5CPU . . . 339
Appendix 4 Use Project in Different Version . . . 340
Appendix 5 Using Simulation Function . . . 342
Supported CPU module functions . . . 343
Supported devices . . . 345
Supported instructions . . . 351
Enabled/disabled parameter items . . . 353
Module buffer memory . . . 354
Link devices of network module . . . 354
Considerations . . . 355
Appendix 6 USB Driver Installation Procedure . . . 357

INDEX **359**

REVISIONS 361
TRADEMARKS 362

RELEVANT MANUALS

Manual name [manual number]	Description	Available form
GX Works3 Operating Manual [SH-081215ENG] (this manual)	Explains the system configuration of GX Works3, parameter setting, and operation method for the online function.	e-Manual EPUB PDF

Point

e-Manual refers to the Mitsubishi FA electronic book manuals that can be browsed using a dedicated tool.
e-Manual has the following features:

- Required information can be cross-searched in multiple manuals.
- Other manuals can be accessed from the links in the manual.
- The hardware specifications of each part can be found from the product figures.
- Pages that users often browse can be bookmarked.

TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
Built-in Ethernet CPU	A generic term for a CPU that has an Ethernet port
Control CPU	A CPU module that controls connected I/O modules and Intelligent function modules. In a multiple CPU system, a CPU module by which a module is controlled can be set for each module.
Direct connection	A connection using a USB/serial/Ethernet port of a CPU module
Execution program	A converted program that can be executed on a CPU module
FB instance	A function block pasted on a sequence program
FX5CPU	A generic term for FX5UCPU and FX5UCCPU
GX Configurator	A generic product name for GX Configurator-AD/DA/SC/CT/TC/TI/FL/PT/AS/QP
GX Developer	A generic product name for SWnD5C-GPPW, SWnD5C-GPPW-A, SWnD5C-GPPW-V, and SWnD5C-GPPW-VA ('n' indicates version.)
GX Works2	A generic product name for SWnDNC-GXW2 ('n' indicates version.)
GX Works3	A generic product name for SWnDND-GXW3 ('n' indicates version.)
Intelligent function module	A generic term for modules which has functions other than input and output, such as A/D converter module and D/A converter module
MELSOFT Library	A generic term for components (FB library, sample programs, drawing data) which provide easy operation of programmable controller related modules and GOTs, and various devices that are connected to them To obtain components of MELSOFT Library, please consult your local Mitsubishi representative.
MELSOFT Navigator	A product name for the integrated development environment included in SWnDND-IQWK (MELSOFT iQ Works) ('n' indicates version.)
Member	A local label defined in a structure or a function block.
Personal computer	A generic term for personal computer on which Windows® operates
RCPU	A generic term for RnCPU, RnENCPU, RnPCPU, and RnSFCPU
RnCPU	A generic term for R04CPU, R08CPU, R16CPU, R32CPU, and R120CPU
RnENCPU	A generic term for R04ENCPU, R08ENCPU, R16ENCPU, R32ENCPU, and R120ENCPU
RnPCPU	A generic term for R08PCPU, R16PCPU, R32PCPU, and R120PCPU
RnSFCPU	A generic term for R08SFCPU, R16SFCPU, R32SFCPU, and R120SFCPU
Slave station	Stations other than master station (local station, remote I/O station, remote device station, intelligent device station, etc.)
Universal model High-speed Type QCPU	A generic term for Q03UDVCPU, Q04UDVCPU, Q06UDVCPU, Q13UDVCPU, and Q26UDVCPU
Universal model QCPU	A generic term for Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q03UDECPU, Q04UDHCPU, Q04UDEHCPU, Q06UDHCPU, Q06UDEHCPU, Q10UDHCPU, Q10UDEHCPU, Q13UDHCPU, Q13UDEHCPU, Q20UDHCPU, Q20UDEHCPU, Q26UDHCPU, Q26UDEHCPU, Q50UDEHCPU, and Q100UDEHCPU
Windows® 7	A generic term for Microsoft® Windows® 7 Starter Operating System, Microsoft® Windows® 7 Home Premium Operating System, Microsoft® Windows® 7 Professional Operating System, Microsoft® Windows® 7 Ultimate Operating System, and Microsoft® Windows® 7 Enterprise Operating System
Windows® 7 or later	A generic term for Windows® 7, Windows® 8, and Windows® 8.1

Term	Description
Windows® 8	A generic term for Microsoft® Windows® 8 Operating System, Microsoft® Windows® 8 Pro Operating System, and Microsoft® Windows® 8 Enterprise Operating System
Windows® 8.1	A generic term for Microsoft® Windows® 8.1 Operating System, Microsoft® Windows® 8.1 Pro Operating System, and Microsoft® Windows® 8.1 Enterprise Operating System
Windows® 8 or later	A generic term for Windows® 8 and Windows® 8.1
Windows Vista®	A generic term for Microsoft® Windows Vista® Home Basic Operating System, Microsoft® Windows Vista® Home Premium Operating System, Microsoft® Windows Vista® Business Operating System, Microsoft® Windows Vista® Ultimate Operating System, and Microsoft® Windows Vista® Enterprise Operating System
Windows Vista® or later	A generic term for Windows Vista®, Windows® 7, Windows® 8, and Windows® 8.1
Windows® XP	A generic term for Microsoft® Windows® XP Home Edition Operating System, and Microsoft® Windows® XP Professional Operating System

MEMO

PART 1 FUNDAMENTALS OF GX Works3

This part explains the screen configuration and basic operations of GX Works3.

1 BEFORE USING THIS PRODUCT

2 SCREEN CONFIGURATION AND BASIC OPERATIONS

1 BEFORE USING THIS PRODUCT

GX Works3 is an engineering tool for configuring the settings, programming, debugging, and performing the maintenance for programmable controllers such as MELSEC iQ-R series/MELSEC iQ-F series.
The usability is much better than the existing GX Works2 because the functions and their operability have been improved.

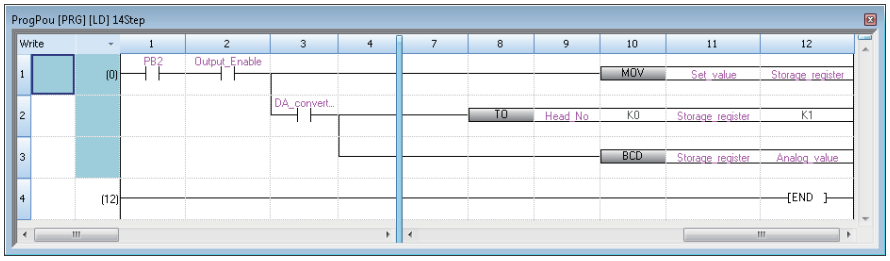
1.1 Main Functions of GX Works3

GX Works3 manages programs and parameters in each project for each CPU module.
GX Works3 has the functions shown below.

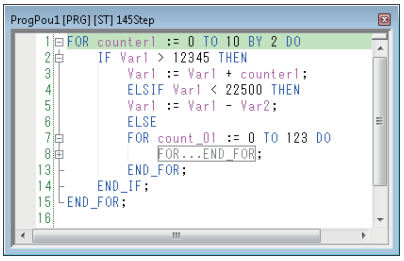
Program creation function

Programs can be created in proper programming languages depending on the processing contents.

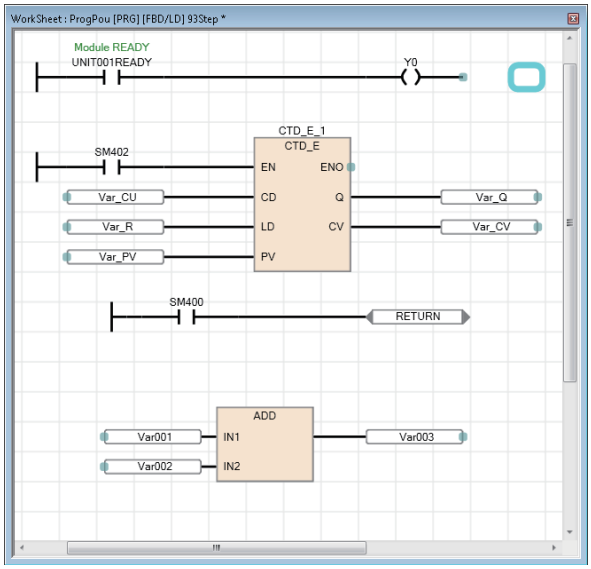
<Ladder program>



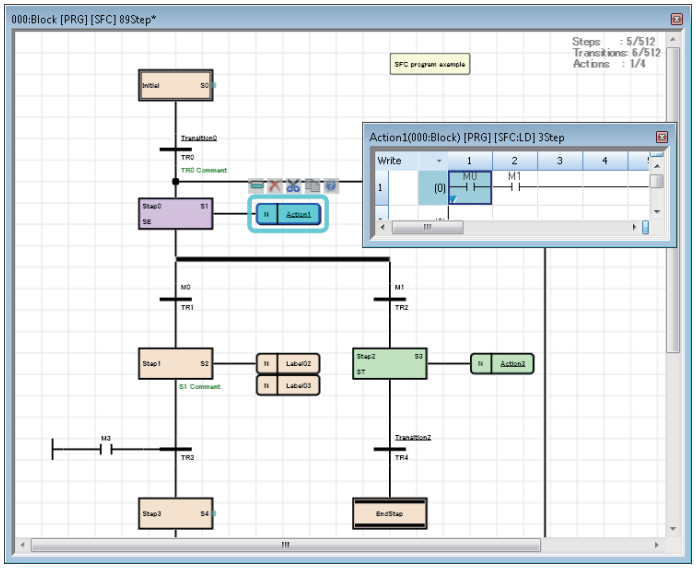
<ST program>



<FBD/LD program>

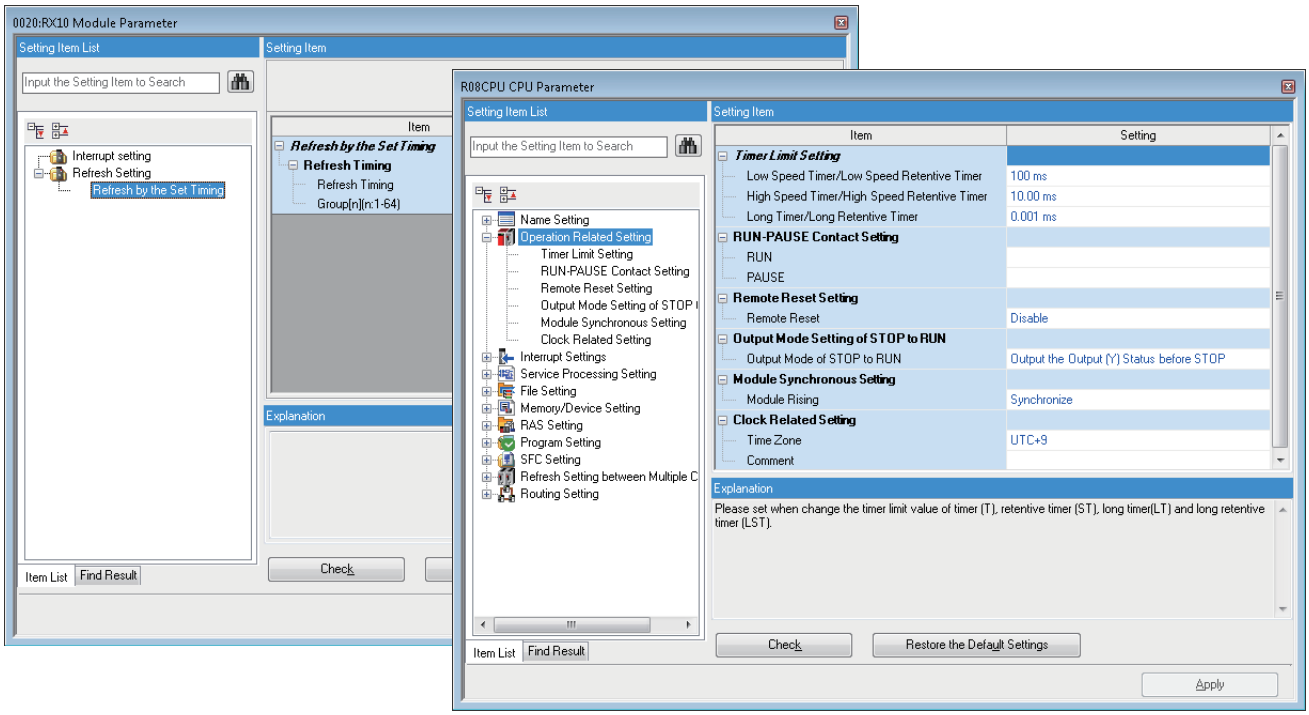


〈SFC program〉



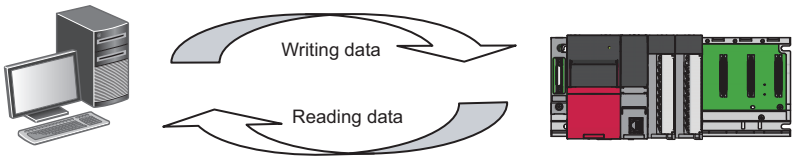
Parameter setting function

Parameters of CPU module, I/O module, and Intelligent function module can be set by using this function.



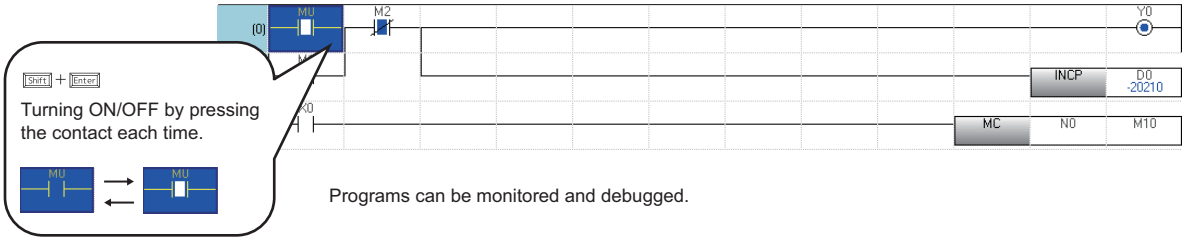
Write to/Read from PLC function

The created sequence programs can be written to/read from a CPU module with the Write to PLC/Read from PLC function. In addition, the sequence programs can be modified while the CPU module is in RUN state by the performing online program change function.



Monitor/debug function

Data such as device values during operation can be monitored by writing a created sequence program to a CPU module. If a CPU module is not connected, programs can be debugged by using the virtual programmable controller (the Simulation function).

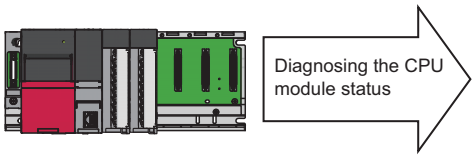


Programs can be monitored and debugged.

Diagnostic function

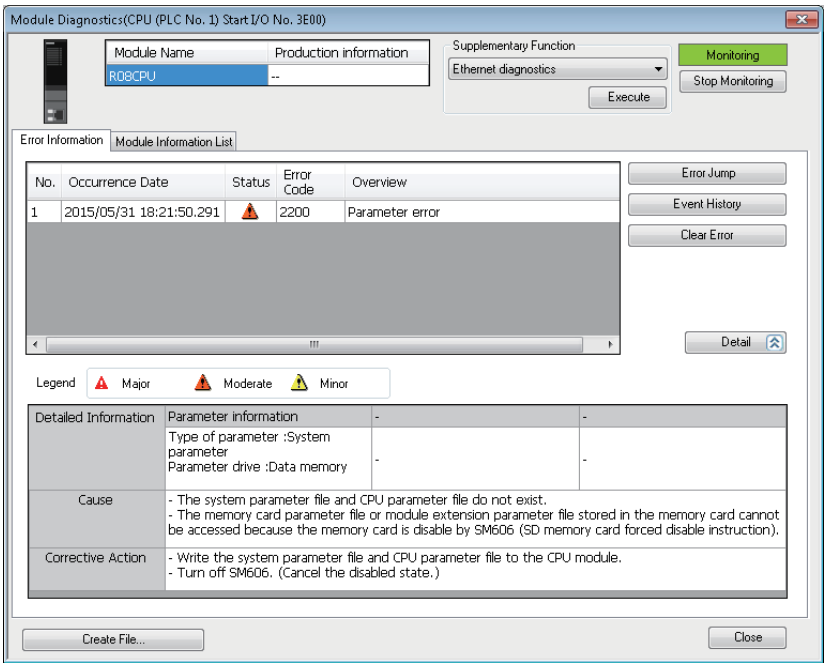
The current error status and the error history of the CPU module or the network can be diagnosed. By using the Diagnostic function, recovery work can be completed in a short time.

The detailed information of modules, such as Intelligent function module, can be obtained by using the System Monitor function. This helps to shorten the recovery time when an error occurs.



Diagnosing the CPU module status

Diagnosing the CPU module ("Module Diagnostics" screen)



1.2 RnSFCPU Supported Functions

GX Works3 is supported by MELSEC iQ-R series RnSFCPU.

In this manual, a project for RnSFCPU is defined as a 'safety project', and a project other than one for RnSFCPU is defined as a 'standard project'.

There are two types of data In a safety project; safety data and standard data.

Both safety programs and standard programs can be created in a safety project.

Name		Description
Data	Safety data	Indicates safety programs, safety FBs/FUNs, safety global labels, and parameters for RnSFCPU.
	Standard data	Indicates standard programs, standard FBs/FUNs, standard global labels, standard/safety shared global labels, and parameters for CPU module other than RnSFCPU.
Program, FB/FUN	Safety program, Safety FB/FUN	A program to execute a safety control.
	Standard program, Standard FB/FUN	A program to execute a general control.
Label	Safety global label	Global labels that can be used only in safety programs.
	Standard global label	Global labels that can be used only in standard programs.
	Standard/safety shared global label	Global labels that can be used in both safety programs and standard programs. To use standard/safety shared global labels in a safety program, create the program so that the safety condition can be checked.

Usable devices, labels, and FB/FUN

Usable devices, labels, and FBs/FUNs differ between a safety project and a standard project.

■Usable devices and labels

Project	Safety device	Safety global label	Standard/safety shared global label	Standard device	Standard global label
Safety program	○	○	○	×	×
Standard program	×	×	○	○	○

For details on usable devices in each project, refer to the following section.

☞ Page 337 Applicable Devices in GX Works3

■Usable FB/FUN

Project	Safety FB	Safety FUN	Standard FB	Standard FUN
Safety program	○	○	×	×
Standard program	×	×	○	○

Safety operation mode of RnSFCPU

■Safety operation mode

There are two types of safety operation mode in RnSFCPU; the safety mode and the test mode.

- Safety mode: A mode used when running a system. This mode prohibits operations which change the control of CPU module such as writing data to the CPU module, changing current values, operating CPU memory, and setting security keys and file passwords.
- Test mode: A mode used when starting the system or performing maintenance. All of the functions can be used. (The available functions differ depending on the access level of the user who is logging on to the personal computer.)

For the restrictions of a safety mode, refer to the following manual.

📖 MELSEC iQ-R CPU Module User's Manual (Application)

For switching the safety operation modes, refer to the following section.

☞ Page 327 Switch Safety Operation Mode

Safety supported function list

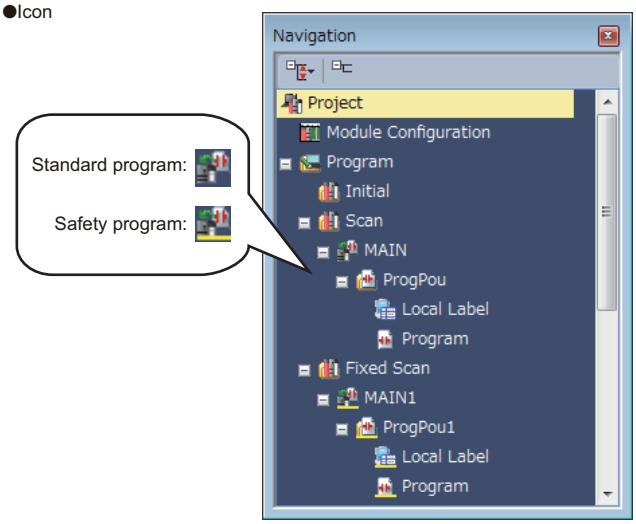
GX Works3 provides the functions to create a safety project.

Purpose	Function name	Reference
To perform user authentication before opening a project in order to prevent illegal access from users with no authority.	User Authentication	Page 303 Preventing Illegal Access to Project
To perform user authentication before accessing a CPU module in order to prevent illegal access from users with no authority.		
To check if the data in RnSFCPU is the one written by the user.	Check Safety Data Identity	Page 329 Check Safety Data Identity
To switch the safety operation mode of RnSFCPU.	Switch Safety Operation Mode	Page 327 Switch Safety Operation Mode

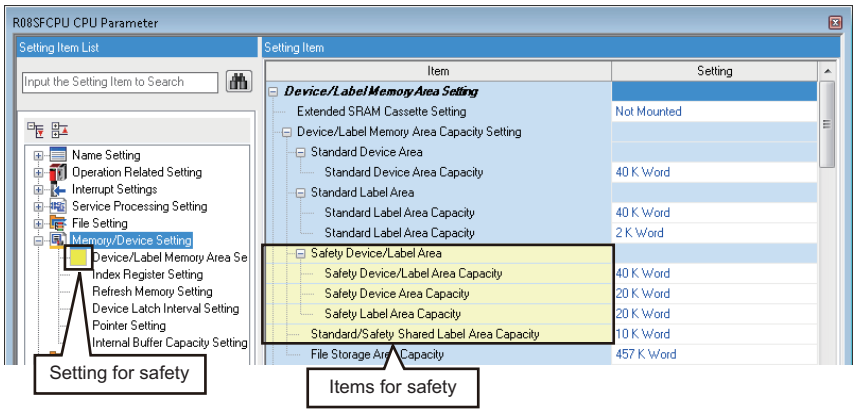
Display of safety data identification

The icons and background colors of safety data are highlighted.

●Icon



●Grid background color



Point

The data category (standard/safety) can be checked on the "Properties" screen for each data. (☞ Page 74 Property)

1.3 Procedure from Project Creation to CPU Module Operation

The operating procedure from program creation to a CPU module operation in GX Works3 is shown below.

For the operation methods of CPU modules, refer to the following manuals.

📖 MELSEC iQ-R CPU Module User's Manual (Application)

📖 MELSEC iQ-F FX5 User's Manual (Application)

Standard project

Start

⇩

1. Create a project.

Start GX Works3.

Create a new project.

☞ Page 27 Start
☞ Page 61 Creating project

⇩

2. Set the parameters.

☞ Page 93 Setting Parameters

⇩

3. Create a program.

Create a Program Organization Unit (POU).

Set an execution order and an execution type.

Set the global labels/local labels.

Edit the program of each POU.

Perform conversion.

☞ Page 72 Creating projects
☞ Page 121 Program execution order/execution type settings
☞ Page 107 Registering Labels
☞ Page 123 Creating Ladder Programs, Page 145 Creating ST Programs, Page 151 Creating FBD/LD Programs, Page 158 Creating SFC Programs, Page 214 Creating Function Blocks, Page 217 Creating Functions
☞ Page 184 Converting Programs

⇩

4. Perform debugging with the simulator.[*1]

☞ Page 226 SIMULATING PROGRAM

*1: FX5CPU does not support it.

⇩

5. Connect a personal computer to a CPU module, and set the connection destination.

☞ Page 235 SETTING ROUTE TO CPU MODULE

⇩

6. Write parameters/programs to the CPU module.

☞ Page 250 Writing data to programmable controller

⇩

7. Check the operation.
Monitor the execution status and device contents of the sequence program to check the operation.
Check the error occurrence in the CPU module.

☞ Page 268 Check Execution Programs on Program Editor
☞ Page 314 Diagnosing CPU Module

⇩

8. Operate the system.

Safety project

Start

⇩

1. Create a project.
Start GX Works3.
Create a new project.
Register user information.
Save the project.

☞ Page 27 Start
☞ Page 61 Creating project
☞ Page 305 User management
☞ Page 69 Saving project

⇩

2. Set the parameters.
Setting the safety device/label area is required.

☞ Page 93 Setting Parameters

⇩

3. Create a program.
Create a Program Organization Unit (POU).
Set an execution order and an execution type.
Set the global labels/local labels.
Edit the program of each POU.
Perform conversion.

☞ Page 72 Creating projects
☞ Page 121 Program execution order/execution type settings
☞ Page 107 Registering Labels
☞ Page 123 Creating Ladder Programs, Page 214 Creating Function Blocks, Page 217 Creating Functions
☞ Page 184 Converting Programs

⇩

4. Connect a personal computer to a CPU module, and set the connection destination.

☞ Page 235 SETTING ROUTE TO CPU MODULE

5. Write parameters/programs to the CPU module.

☞ Page 250 Writing data to programmable controller

6. Check the operation. (Test mode)
Monitor the execution status and device contents of the sequence program to check the operation.
Check the error occurrence in the CPU module.

☞ Page 268 Check Execution Programs on Program Editor
☞ Page 314 Diagnosing CPU Module

7. Switch the safety operation modes.
Check whether the written project is correct.
Switch it the safety mode.

☞ Page 329 Check Safety Data Identity
☞ Page 327 Switch Safety Operation Mode

8. Operate the system.

1.4 Learning Operation Method of GX Works3

This section explains how to learn the operation methods of GX Works3.

Displaying Help

Open the Help to learn about operations and functions, and check the error codes of a CPU module.

Operating procedure

Select [Help] ⇨ [GX Works3 Help] ( ).
e-Manual Viewer starts and a manual is displayed.

Find Help function

Enter a character string in on the toolbar and press the Enter key to start searching in the e-Manual Viewer.
The entered character string can be searched in the manuals which have been registered in the e-Manual Viewer.

Help of instructions/FBs and special relays/special registers

The corresponding location in the manual can be displayed from each program editor or screen.

■Program editor

- Place the cursor on the instruction (ladder editor: the cell with the instruction, ST editor: the character string of the instruction), press the F1 key.
- Place the cursor on the element on the FBD/LD editor, and press the F1 key.
- Place the cursor on the FB cell on the ladder editor, and press the F1 key
- Place the cursor on the element on the SFC diagram editor, and press the F1 key.

■"Enter Ladder" screen and argument editing screen

Click the [Manual] button on each screen.

■Element Selection window

Select the instruction or the module FB in the list, and press the F1 key.

Connecting to MITSUBISHI ELECTRIC FA Global Website

Open the MITSUBISHI ELECTRIC FA Global Website in a Web browser.
Make sure your personal computer connect to Internet in advance.

Operating procedure

Select [Help] ⇨ [Connection to MITSUBISHI ELECTRIC FA Global Website].

Checking version of GX Works3

Display the information such as the software version of GX Works3.

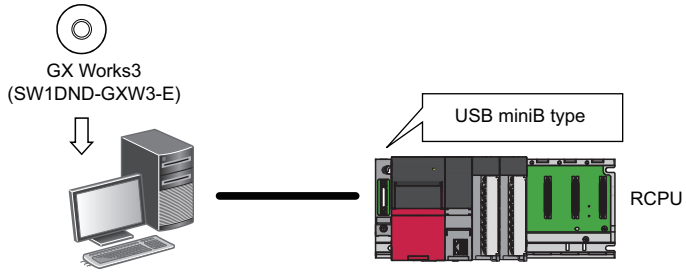
Operating procedure

Select [Help] ⇨ [Version Information].

1.5 Connection Configurations between Personal Computer and CPU Module

The configurations when connecting a personal computer and a CPU module are shown below.

Connection via USB port



FX5CPU does not support this connection.

The USB cables of which the operation is ensured by Mitsubishi Electric Corporation are shown below.

When using a USB cable for the first time, install the USB driver.

For details, refer to the following section.

☞ Page 357 USB Driver Installation Procedure

Product name	Model	Manufacturer
USB cable (USB A type — USB B type)	AU230	BUFFALO KOKUYO SUPPLY INC.
USB cable (USB A type — USB miniB type)	KU-AMB530	SANWA SUPPLY INC.
	KU-AMB550	
USB adapter (USB B type — USB miniB type)	AD-USBBFTM5M	ELECOM Co., Ltd.

For any considerations when accessing a CPU module, refer to the following section.

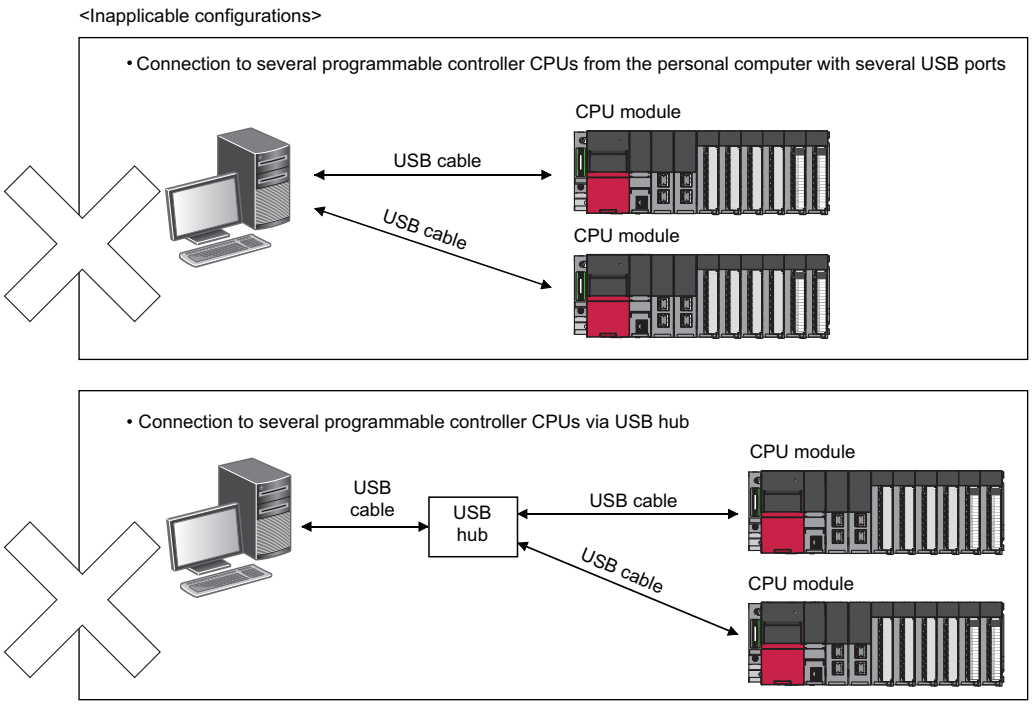
☞ Page 247 Considerations of communication with the CPU module using a USB cable

■Configuration of USB connection

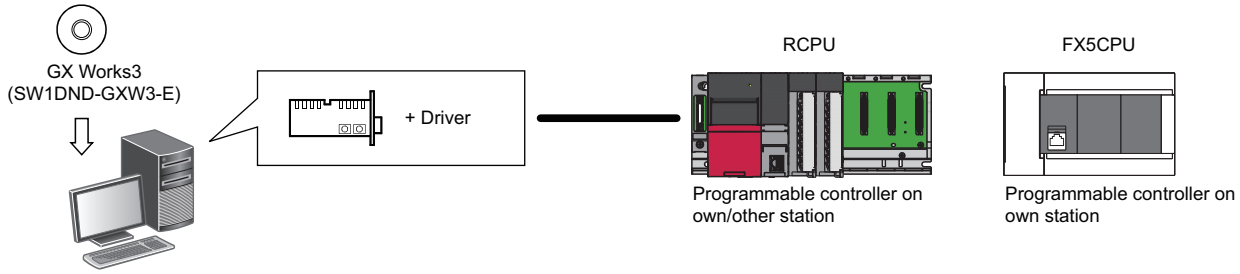
Only one CPU module can be connected to a personal computer at the same time.

Connection to a CPU module is not applicable for the following configuration.

<Inapplicable configurations>

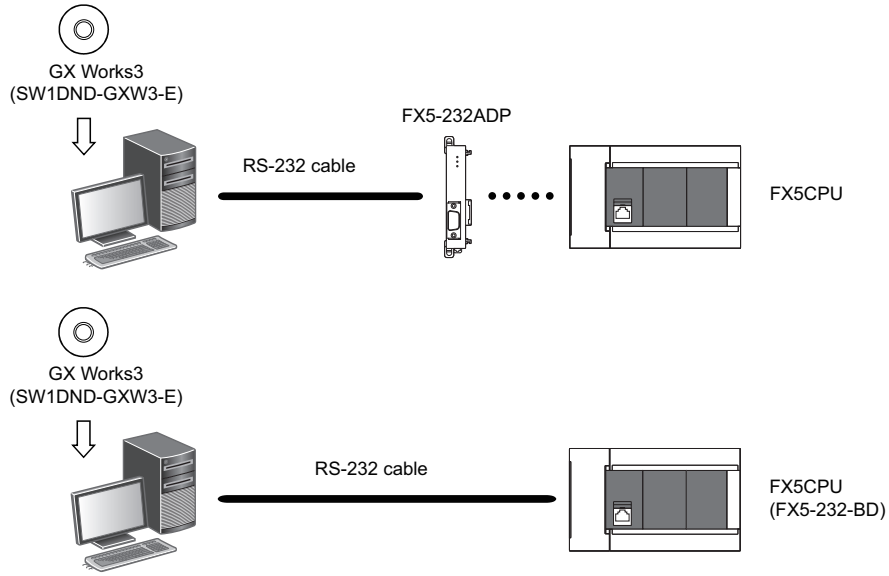


Connection via Ethernet board



Use an Ethernet board that is built-in to a personal computer or commercially available.

Connection via serial port



RCPU does not support this connection.

Product name	Model	Manufacturer
RS-232 cable	FX-232CAB-1	Mitsubishi Electric Corporation

2 SCREEN CONFIGURATION AND BASIC OPERATIONS

This chapter explains the screen configuration and basic operations of GX Works3.

2.1 Starting and Exiting GX Works3

This section explains the operation methods for starting and exiting GX Works3.

Start

Operating procedure

Select [MELSOFT] ⇨ [GX Works3] ⇨ [GX Works3] from Windows® Start*1.

*1 Select [All apps] on the Start screen or [Start] ⇨ [All Programs].

Exit

Operating procedure

Select [Project] ⇨ [Exit GX Works3].

Point

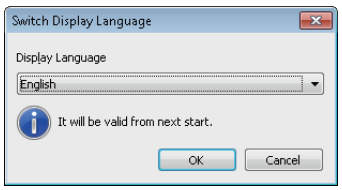
GX Works3 can also be started/exited from MELSOFT Navigator.

2.2 Switching Displayed Language

Since GX Works3 supports displaying in multiple languages, switching the language, such as on the menu bar, is available on a personal computer.

Window

[View] ⇨ [Switch Display Language]



Precautions

If the display language differs from the one for the operating system, the text may get cut off.

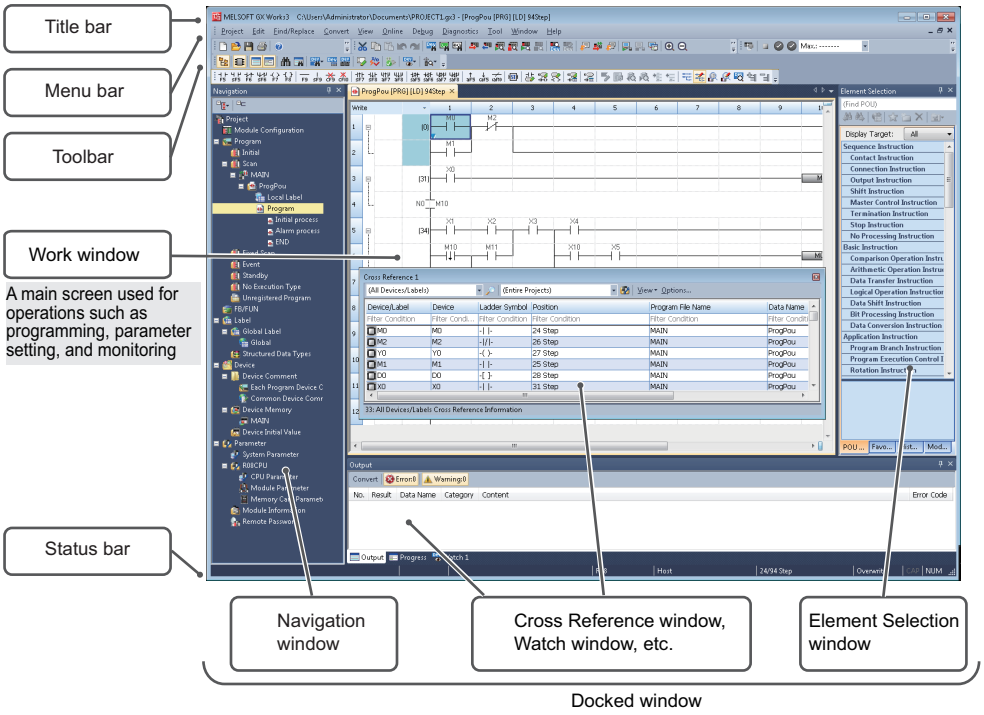
2.3 Screen Configuration

This section explains the screen configuration that is displayed of GX Works3 starts up.

Main frame

The main frame configuration is shown below.

The following screen shows a main frame configuration on which a work window and docked windows are displayed.



Window operation

■Displaying docked windows

[View] ⇨ [Docking Window] ⇨ [(target item)]

Point

When the docked window is not displayed by selecting it from the menu, select [Window] ⇨ [Reset Window Layout Back to Initial Status].

■Switching docked window and work window

The windows or files can be switched by pressing the Ctrl + Tab keys.

Select a particular window or file by pressing the Ctrl + ← / → / ↑ / ↓ keys.

■Arranging work windows

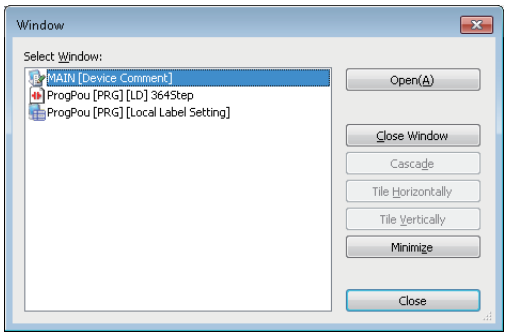
The list of open windows is displayed.

Open and arrange the specific window.

Arranging the windows improves display efficiency while opening multiple windows.

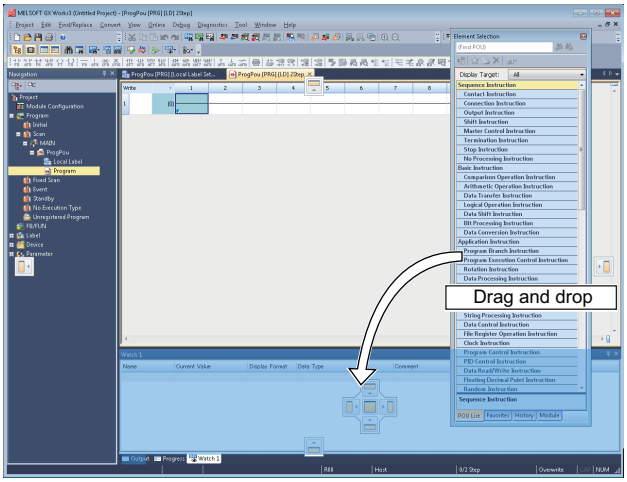
Window

[Window] ⇨ [Window]

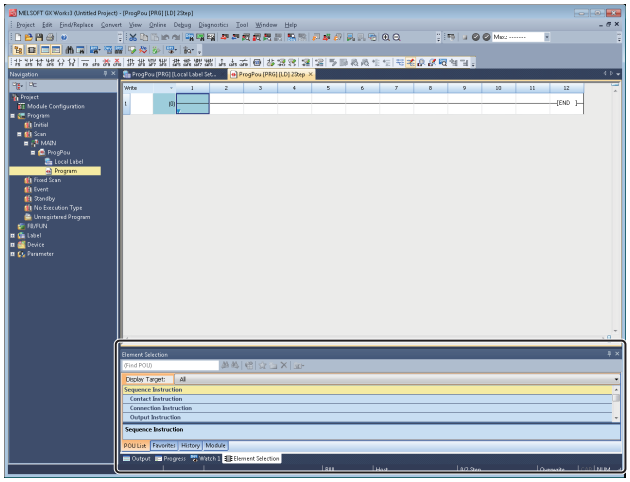


■Docking/floating dockable windows

- Docked display: Drag the title bar of a floating dockable window and drop it to the guidance to dock the main frame.



Drag a dockable window to the guidance



A new tab appears after the window is docked.

- Floating display: Drag the title bar of a docked window and drop it to the arbitrary position to float from the main frame.

■Docking/floating work windows

- Docked display: Select the floating work window and select [Window] ⇨ [Docking].
- Floating display: Select the docked work window and select [Window] ⇨ [Floating].

Point

Docked windows can be switched between the docked display and the floating display by double-clicking the title bar.

Customizing/resetting toolbars

Set the types of tool buttons to be displayed on each toolbar.

The selected tool buttons on the list are displayed on the tool bar.

Operating procedure

■Customizing toolbar

1. Click on the toolbar, and select [Show/Hide Buttons] ⇨ [(toolbar name)].
2. Select the tool button to be displayed from the list.

■Resetting toolbar

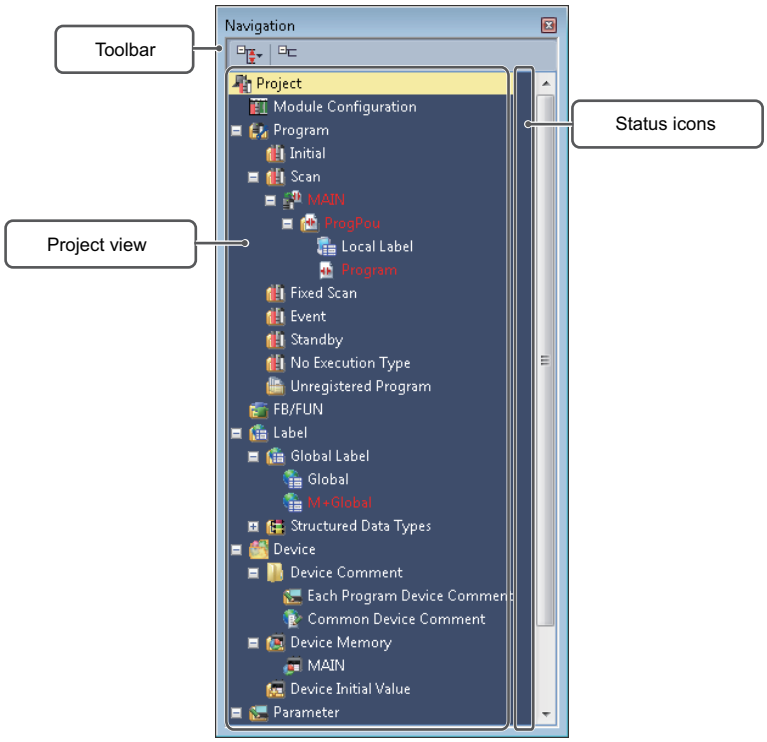
Click on the toolbar, and select [Show/Hide Buttons] ⇨ [Reset].

Navigation window

Navigation window is the screen that displays the content of a project in a tree format.
Operations such as creating new data and displaying editing screens can be performed on the tree.
For details, refer to the following section.
☞ Page 56 PROJECT MANAGEMENT

Window

[View] ⇨ [Docking Window] ⇨ [Navigation] ( )



The character color on the Project view displays red when the program is not converted.

Displayed items

Name	Description	Reference
Status icons	Displays icons which indicate status of the project.	Page 32 Status icons

Sorting data

Sort the data displayed in tree format.

Operating procedure

Select a program file, then right-click and select [Sort] ⇨ [(sort type)] from the shortcut menu.

Point

Data can be sorted by dragging and dropping the data or selecting [Order] ⇨ [Move Up]/[Move Down] from the shortcut menu.

Creating folders

Create a folder for grouping and managing the created data.

Operating procedure

1. Select a program file and select [Project] ⇨ [Data Operation] ⇨ [New Folder]/right-click and select [New Folder] from the shortcut menu.
2. Change a folder name.
3. Select the program to be stored, and drag and drop it onto the created folder.

Simple display

Click on the toolbar to hide the unused folders.

Status icons

The icons that indicate the state of the project are as follows:

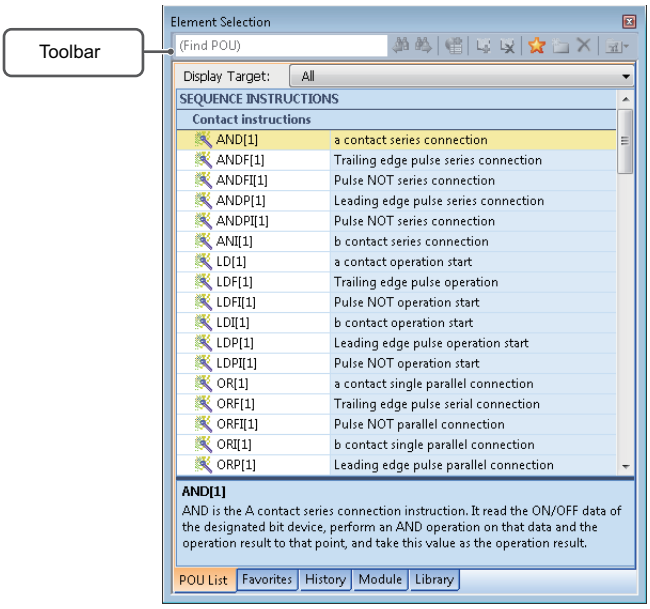
Icon	Status	Display timing	Item	Description
	Parameter mismatched	Offline	Module folder	Displays when a mismatch was detected between the system parameters and the property of module.
	Unconfirmed required settings		Module parameter	Displays when the [Apply] button has never been pressed on the setting screen of module parameter (network) that is required to be set.

Element Selection window

The Element Selection window is the window that displays elements (such as instructions and standard functions/function blocks) for program creation in a list format.

Window

[View] ⇨ [Docking Window] ⇨ [Element Selection] ( )


Toolbar
Element Selection
(Find POU)
Display Target:
All
SEQUENCE INSTRUCTIONS
Contact instructions
AND[1] a contact series connection
ANDF[1] Trailing edge pulse series connection
ANDFI[1] Pulse NOT series connection
ANDP[1] Leading edge pulse series connection
ANDPI[1] Pulse NOT series connection
ANI[1] b contact series connection
LD[1] a contact operation start
LDF[1] Trailing edge pulse operation
LDFI[1] Pulse NOT operation start
LDI[1] b contact operation start
LDP[1] Leading edge pulse operation start
LDPI[1] Pulse NOT operation start
OR[1] a contact single parallel connection
ORF[1] Trailing edge pulse serial connection
ORFI[1] Pulse NOT parallel connection
ORI[1] b contact single parallel connection
ORP[1] Leading edge pulse parallel connection
AND[1]
AND is the A contact series connection instruction. It read the ON/OFF data of the designated bit device, perform an AND operation on that data and the operation result to that point, and take this value as the operation result.
POU List
Favorites
History
Module
Library

- The focus is moved to the element of which character string match with the entered character string (element name or keyword included in the element description) to the toolbar.
- The elements which are included in the selected category is displayed from the display target.

Pasting elements

■Pasting elements in a program

When the program editor is activated, the elements which can be pasted are displayed in the Element Selection window.
The elements can be pasted in the program editor by dragging and dropping from the list to a program.

Point

For ST editor, select an element and press the Enter key to paste the element at the cursor position.

■Pasting objects in Module Configuration Diagram

When the Module Configuration Diagram is activated, the module parts (objects) which can be pasted are displayed in the Element Selection window.
The objects can be pasted by dragging and dropping from the list to the Module Configuration Diagram.

Favorites

The frequently used elements/SFC elements (devices, labels, instructions, FB instances, functions) can be registered in the [Favorites] tab by classification.
Create a folder to be classified by clicking on the toolbar, and drag and drop it to move or change the name.

Operating procedure

■Adding from element list

Instructions, functions, and function blocks can be added to Favorites.

1. Select an element to be added from the list of the Element Selection window, and click on the toolbar.
2. Select the registration destination on the "Register to Favorites" screen, and click the [OK] button.

■Adding from the Project view

Functions and function blocks can be added to Favorites.

1. Select an element to be added on the Project view, and drag and drop it onto the Element Selection window.
2. Select the registration destination on the "Register to Favorites" screen, and click the [OK] button.

■Adding from ladder editor

Devices, labels, instructions, and FB instances can be added to Favorites.

1. Select the cell of the element to be added, and drag the border of the cell and drop it onto the Element Selection window.
2. Select the registration destination on the "Register to Favorites" screen, and click the [OK] button.

■Adding from ST editor

Devices, labels, and FB instances can be added to Favorites.

1. Select the token of a part to be added, and drag and drop it onto the Element Selection window.
2. Select the registration destination on the "Register to Favorites" screen, and click the [OK] button.

■Adding from FBD/LD editor

Devices, labels, and FB instances can be added to Favorites.

1. Select the element to be added, and drag and drop it onto the Element Selection window with the Ctrl key held down.
2. Select the registration destination on the "Register to Favorites" screen, and click the [OK] button.

■Adding from SFC diagram editor

Devices and labels can be added to Favorites.

1. Select the SFC element to be added, and drag and drop it onto the Element Selection window.
2. Select the registration destination on the "Register to Favorites" screen, and click the [OK] button.

■Adding from label editor

Labels can be added to Favorites.

1. Select a line header of the label to be added, and drag and drop it onto the Element Selection window.
2. Select the registration destination on the "Register to Favorites" screen, and click the [OK] button.

■Adding templates classified by purpose

GX Works3 provides templates ("Category by Target Template") that are categorized by purpose to which register the arbitrary elements as a favorite in batch.

By deleting unnecessary elements after registering the templates, the elements can be placed on efficiency.

1. Select the [Favorites] tab on the Element Selection window.
2. Select from the tool bar ⇨ [Import Favorites] ⇨ [Category by Target Template] in the Element Selection window.

Point

By importing the exported file (*.xml), the elements registered as favorite can be used on other personal computer.

Select on the toolbar ⇨ [Export Favorites]/[Import Favorites]

History

By selecting the [History] tab, the previously used elements are displayed in order of the date used.

The display order can be changed in order of the number of times used from the pulldown list.

Module

By selecting the [Module] tab, the module labels and the module FBs which have been registered in the project are displayed.

For details on how to register module labels and module FBs, refer to the following section.

☞ Page 115 Registering Module Labels, Page 216 Importing module FBs in project

Library

By selecting the [Library] tab, the POUs which have been registered in the library file are displayed.

For details on how to register POUs, refer to the following section.

☞ Page 221 Registering to library list

2.4 Menu List

Basic menus

[Project]	
⇨ [New]	Page 61 Creating project
⇨ [Open]	Page 63 Opening project
⇨ [Close]	—
⇨ [Save]	Page 69 Overwriting projects
⇨ [Save As]	Page 69 Saving projects under the specified name
⇨ [Delete]	Page 70 Deleting project
⇨ [Project Verify]	Page 77 Verifying Projects
⇨ [Project Revision] ⇨ [Register Revision]	Page 80 Registering revision information
⇨ [Project Revision] ⇨ [Revision List]	Page 81 Displaying revision list
⇨ [Change Module Type]	Page 75 Changing Module Type of Project
⇨ [Data Operation] ⇨ [New Data]	Page 72 Creating projects
⇨ [Data Operation] ⇨ [New Folder]	Page 32 Creating folders
⇨ [Data Operation] ⇨ [Rename Data]	Page 73 Changing a data name
⇨ [Data Operation] ⇨ [Delete Data]	Page 73 Deleting data
⇨ [Data Operation] ⇨ [Copy Data]	Page 73 Copying/pasting data
⇨ [Data Operation] ⇨ [Paste Data]	
⇨ [Data Operation] ⇨ [Add New Module]	Page 97 Setting parameters of I/O module and Intelligent function module Page 103 Simple motion module setting tool
⇨ [Data Operation] ⇨ [Property]	Page 74 Property
⇨ [Intelligent Function Module] ⇨ [Module Parameter List]	Page 99 Checking/changing the number of Intelligent function module parameters
⇨ [Open Other Format File] ⇨ [GX Works2 Format] ⇨ [Open Project]	Page 64 Opening GX Works2 format project
⇨ [Open Other Format File] ⇨ [GX Works2 Format] ⇨ [Open User Library]	Page 220 Creating a library in GX Works3 format from a library in GX Works2 format
⇨ [Open Other Format File] ⇨ [GX Works3 Format] ⇨ [Open User Library]	Page 221 Editing library
⇨ [Library Operation] ⇨ [Export Library]	Page 220 Creating library file
⇨ [Library Operation] ⇨ [Register to Library List]	Page 221 Registering library to be referenced
⇨ [Library Operation] ⇨ [Delete from Library List]	Page 222 Deleting library/updating display information
⇨ [Library Operation] ⇨ [Update the Display Information of Library]	
⇨ [Security] ⇨ [User Management]	Page 305 User management
⇨ [Security] ⇨ [Change User Password]	Page 304 Changing password of logon user
⇨ [Security] ⇨ [Security Key Setting]	Page 299 Registering security key in program file Page 301 Writing/deleting security key to/from CPU module
⇨ [Security] ⇨ [Security Key Management]	Page 297 Creating/deleting security key
⇨ [Security] ⇨ [Block Password Setting]	Page 293 Setting block password
⇨ [Security] ⇨ [File Password Setting]	Page 309 Setting file password
⇨ [Printer Setup]	Page 51 Printing Data
⇨ [Page Setup]	
⇨ [Print Preview]	
⇨ [Print]	
⇨ [Recent Projects] ⇨ [(recently used project 1 to 10)]	—
⇨ [Start GX Works2]	
⇨ [Exit GX Works3]	Page 27 Exit

[Edit]	
⇨ [Undo]	—
⇨ [Redo]	
⇨ [Cut]	
⇨ [Copy]	
⇨ [Paste]	
[Find/Replace]	
⇨ [Cross Reference]	Page 210 Creating/displaying cross reference information
⇨ [Device List]	Page 212 Displaying Device Usage List
⇨ [Find Device/Label]	Page 202 Searching/Replacing Devices/Labels
⇨ [Find Instruction]	Page 204 Searching/Replacing Instructions
⇨ [Find Contact or Coil]	
⇨ [Find String]	Page 205 Searching/Replacing Character Strings
⇨ [Replace Device/Label]	Page 202 Searching/Replacing Devices/Labels
⇨ [Replace Instruction]	Page 204 Searching/Replacing Instructions
⇨ [Replace String]	Page 205 Searching/Replacing Character Strings
⇨ [Change Open/Close Contact]	Page 207 Changing Contacts Between Open Contact and Close Contact
⇨ [Device Batch Replace]	Page 208 Batch Replacing of Devices and Labels
⇨ [Register to Device Batch Replace]	—
[Convert]	
⇨ [Convert]	Page 184 Converting/converting all programs
⇨ [Online Program Change]	Page 259 Writing data after changing programs partially while CPU module is in RUN (Online program change)
⇨ [Rebuild All]	Page 184 Converting/converting all programs
⇨ [Program File Setting]	Page 121 Program execution order settings
⇨ [Setting]	—
[View]	
⇨ [Toolbar] ⇨ [Standard]	Page 28 Main frame
⇨ [Toolbar] ⇨ [Program Common]	
⇨ [Toolbar] ⇨ [Docking Window]	
⇨ [Toolbar] ⇨ [Monitor Status]	
⇨ [Statusbar]	
⇨ [Color and Font]	Page 48 Checking and Changing Colors and Fonts
⇨ [Docking Window] ⇨ [Navigation]	Page 31 Navigation window
⇨ [Docking Window] ⇨ [Element Selection]	Page 32 Element Selection window
⇨ [Docking Window] ⇨ [Output]	Page 28 Main frame
⇨ [Docking Window] ⇨ [Progress]	
⇨ [Docking Window] ⇨ [Find/Replace]	—
⇨ [Docking Window] ⇨ [Find Results]	
⇨ [Docking Window] ⇨ [(Cross Reference 1 to 2)]	Page 210 Creating/displaying cross reference information
⇨ [Docking Window] ⇨ [Device List]	Page 212 Displaying Device Usage List
⇨ [Docking Window] ⇨ [Device Reference]	Page 98 Checking refresh devices assigned to modules
⇨ [Docking Window] ⇨ [Input the Configuration Detailed Information]	—
⇨ [Docking Window] ⇨ [Result of Power Supply Capacity and I/O Points Check]	
⇨ [Docking Window] ⇨ [Module Start I/O No. Related Area]	
⇨ [Docking Window] ⇨ [(Watch 1 to 4)]	Page 277 Checking Current Values by Registering Devices/Labels
⇨ [Docking Window] ⇨ [Intelligent Function Module Monitor] ⇨ [(Intelligent Function Module Monitor 1 to 10)]	Page 282 Checking Current Values in Intelligent Function Module

[View]	
⇨ [Zoom] ⇨ [Set Zoom Factor]	—
⇨ [Zoom] ⇨ [Zoom In]	
⇨ [Zoom] ⇨ [Zoom Out]	
⇨ [Switch Display Language]	Page 27 Switching Displayed Language
⇨ [Multiple Comment Display Setting]	Page 49 Setting Comment Display
[Online]	
⇨ [Specify Connection Destination]	Page 235 Specification of Connection Destination
⇨ [Read from PLC]	Page 248 Writing/Reading Programmable Controller Data
⇨ [Write to PLC]	
⇨ [Verify with PLC]	Page 257 Verifying Programmable Controller Data
⇨ [Remote Operation]	Page 326 CPU Module Remote Operation
⇨ [Safety PLC Operation] ⇨ [Check Safety Data Identity]	Page 329 Check Safety Data Identity
⇨ [Safety PLC Operation] ⇨ [Switch Safety Operation Mode]	Page 327 Switch Safety Operation Mode
⇨ [CPU Memory Operation]	Page 330 Checking Memory Usage
⇨ [Delete PLC Data]	Page 248 Configuration of Online Data Operation screen
⇨ [User Data] ⇨ [Read]	Page 262 Reading/Writing/Deleting User Data
⇨ [User Data] ⇨ [Write]	
⇨ [User Data] ⇨ [Delete]	
⇨ [Set Clock]	Page 325 Setting Clock in CPU Module
⇨ [Monitor] ⇨ [Monitor Mode]	Page 126 Read mode/Write mode/Monitor read mode/ Monitor write mode
⇨ [Monitor] ⇨ [Monitor (Write Mode)]	
⇨ [Monitor] ⇨ [Start Monitoring (All Windows)]	Page 266 Starting/stopping monitoring
⇨ [Monitor] ⇨ [Stop Monitoring (All Windows)]	
⇨ [Monitor] ⇨ [Start Monitoring]	
⇨ [Monitor] ⇨ [Stop Monitoring]	
⇨ [Monitor] ⇨ [Change Value Format (Decimal)]	Page 266 Changing display format of word devices
⇨ [Monitor] ⇨ [Change Value Format (Hexadecimal)]	
⇨ [Monitor] ⇨ [Device/Buffer Memory Batch Monitor]	Page 275 Checking Device/Buffer Memory in Batch
⇨ [Monitor] ⇨ [Program List Monitor]	Page 280 Checking Processing Time of Program
⇨ [Monitor] ⇨ [Interrupt Program List Monitor]	Page 281 Checking Execution Counts of Interrupt Programs
⇨ [Monitor] ⇨ [SFC All Blocks Batch Monitoring]	Page 274 Monitoring all SFC blocks in batch/active steps
⇨ [Watch] ⇨ [Start Watching]	Page 277 Checking Current Values by Registering Devices/Labels
⇨ [Watch] ⇨ [Stop Watching]	
⇨ [Watch] ⇨ [Register to Watch Window] ⇨ [(watch window 1 to 4)]	
⇨ [User Authentication] ⇨ [Log on to PLC]	Page 306 Logging on to CPU module
⇨ [User Authentication] ⇨ [Change the Password of PLC]	Page 306 Changing password
⇨ [User Authentication] ⇨ [Read User Data from PLC]	Page 306 Writing/reading user information to/from CPU module
⇨ [User Authentication] ⇨ [Write User Data to PLC]	
⇨ [User Authentication] ⇨ [Initialize all PLC Data]	Page 306 When a user name and a password used in the User Authentication function are lost
[Debug]	
⇨ [Simulation] ⇨ [Start Simulation]	Page 227 Starting/Stopping Simulation
⇨ [Simulation] ⇨ [Stop Simulation]	
⇨ [Simulation] ⇨ [Simulation Environmental Setting]	Page 227 Environmental setting
⇨ [Modify Value]	Page 265 Changing current values
⇨ [Change History of Current Value]	
⇨ [Memory Dump] ⇨ [Setting]	Page 284 Saving and Checking Device Value
⇨ [Memory Dump] ⇨ [Read Results]	
⇨ [Memory Dump] ⇨ [Display Result]	
⇨ [Offline Monitor]	Page 288 Checking Sampled Data on Program Editor

[Diagnostics]	
⇨ [System Monitor]	Page 313 Checking Module Status of System
⇨ [Module Diagnostics (CPU Diagnostics)]	Page 314 Diagnosing CPU Module
⇨ [Ethernet Diagnostics]	Page 316 Ethernet diagnostics
⇨ [CC-Link IE Control Diagnostics (Optical Cable)]	Page 317 CC-Link IE Controller Network diagnostics
⇨ [CC-Link IE Control Diagnostics (Twisted Pair Cable)]	
⇨ [CC-Link IE Field Diagnostics]	Page 319 CC-Link IE Field Network diagnostics
⇨ [MELSECNET Diagnostics]	Page 320 MELSECNET diagnostics
⇨ [CC-Link Diagnostics]	Page 321 CC-Link diagnostics

[Tool]	
⇨ [Memory Card] ⇨ [Read from Memory Card]	Page 264 Writing to/Reading from memory card
⇨ [Memory Card] ⇨ [Write to Memory Card]	
⇨ [Check Program]	Page 183 Checking Programs
⇨ [Check Parameter]	Page 94 Checking parameters
⇨ [Confirm Memory Size (Offline)]	Page 190 Calculating Memory Size
⇨ [Module Tool List]	Page 103 Displaying module tool list
⇨ [Profile Management] ⇨ [Register]	Page 83 Registration
⇨ [Profile Management] ⇨ [Delete]	Page 83 Deletion
⇨ [Sample Library Registration]	Page 223 Enhanced use of libraries
⇨ [Shortcut Key]	Page 47 Checking and Changing Shortcut Keys
⇨ [Predefined Protocol Support Function]	Page 104 Predefined Protocol Support Function
⇨ [Circuit Trace]	Page 105 Circuit Trace Function
⇨ [Options]	Page 50 Setting Options

[Window]	
⇨ [Cascade]	—
⇨ [Tile Vertically]	
⇨ [Tile Horizontally]	
⇨ [Arrange Icons]	
⇨ [Close All Windows]	
⇨ [Reset Window Layout Back to Initial Status]	Page 28 Displaying docked windows
⇨ [Split]	—
⇨ [Restore Split]	
⇨ [Floating]	Page 29 Docking/floating work windows
⇨ [Docking]	
⇨ [(Window information being displayed)]	—
⇨ [Window]	Page 29 Arranging work windows

[Help]	
⇨ [GX Works3 Help]	Page 24 Displaying Help
⇨ [Connection to MITSUBISHI ELECTRIC FA Global Website]	Page 24 Connecting to MITSUBISHI ELECTRIC FA Global Website
⇨ [Version Information]	Page 24 Checking version of GX Works3

Applicable menus for Module Configuration Diagram

[Edit]	
⇨ [Delete]	—
⇨ [Select All]	
⇨ [Bring to Front]	
⇨ [Send to Back]	
⇨ [Module Status Setting (Empty)]	Page 87 Module status setting (empty)
⇨ [Display Module Information]	Page 88 Check model names on module objects
⇨ [Check] ⇨ [Power Supply Capacity and I/O Points]	Page 92 Checking power supply capacity and I/O points
⇨ [Check] ⇨ [System Configuration]	Page 92 Checking system configurations

[Edit]	
⇨ [Parameter] ⇨ [Fix]	Page 89 Setting parameters on the Module Configuration Diagram
⇨ [Parameter] ⇨ [Input Detailed Configuration Information Window]	—
⇨ [Start XY Batch Input]	Page 92 Inputting start XY in batch
⇨ [Default Points Batch Input]	Page 92 Inputting default points in batch
[View]	
⇨ [Toolbar] ⇨ [Module Configuration Diagram]	—
[Online]	
⇨ [Read Module Configuration from PLC]	Page 87 Reading module configuration from actual system

Applicable menus for parameter editor

[Edit]	
⇨ [Set Maximum Value]	The menus to be displayed differ depending on the module.
⇨ [Set Minimum Value]	
⇨ [Copy Positioning Data]	
⇨ [Paste Positioning Data]	
⇨ [Channel Copy]	
⇨ [Copy Axis]	
⇨ [Refresh Batch Setting] ⇨ [Enable All]	
⇨ [Refresh Batch Setting] ⇨ [Disable All]	
⇨ [Refresh Batch Setting] ⇨ [Back to User Default]	
⇨ [Auto-assignment]	
⇨ [Setting Method] ⇨ [Start/End]	
⇨ [Setting Method] ⇨ [Points/Start]	
⇨ [Device Assignment Method] ⇨ [Start/End]	
⇨ [Device Assignment Method] ⇨ [Points/Start]	
⇨ [Word Device Setting Value Input Format] ⇨ [Decimal]	
⇨ [Word Device Setting Value Input Format] ⇨ [Hexadecimal]	
⇨ [IP Address Input Format] ⇨ [Decimal]	
⇨ [IP Address Input Format] ⇨ [Hexadecimal]	

Applicable menus for ladder editor

[Edit]	
⇨ [Continuous Paste]	Page 140 Pasting device number/label name consecutively
⇨ [Insert and Paste]	Page 139 Pasting
⇨ [Delete]	—
⇨ [Insert Row]	
⇨ [Delete Row]	
⇨ [Insert Column]	
⇨ [Delete Column]	
⇨ [NOP Batch Insert]	Page 139 Inserting/deleting NOP instruction
⇨ [NOP Batch Delete]	
⇨ [Ladder Edit Mode] ⇨ [Read Mode]	Page 126 Read mode/Write mode/Monitor read mode/Monitor write mode
⇨ [Ladder Edit Mode] ⇨ [Write Mode]	

[Edit]	
⇨ [Ladder Symbol] ⇨ [Open Contact]	Page 127 Inserting from the menu/toolbar
⇨ [Ladder Symbol] ⇨ [Close Contact]	
⇨ [Ladder Symbol] ⇨ [Open Branch]	
⇨ [Ladder Symbol] ⇨ [Close Branch]	
⇨ [Ladder Symbol] ⇨ [Coil]	
⇨ [Ladder Symbol] ⇨ [Application Instruction]	
⇨ [Ladder Symbol] ⇨ [Vertical Line]	
⇨ [Ladder Symbol] ⇨ [Horizontal Line]	
⇨ [Ladder Symbol] ⇨ [Delete Vertical Line]	
⇨ [Ladder Symbol] ⇨ [Delete Horizontal Line]	
⇨ [Ladder Symbol] ⇨ [Pulse Contact Symbol] ⇨ [Rising Pulse]	
⇨ [Ladder Symbol] ⇨ [Pulse Contact Symbol] ⇨ [Falling Pulse]	
⇨ [Ladder Symbol] ⇨ [Pulse Contact Symbol] ⇨ [Rising Pulse Branch]	
⇨ [Ladder Symbol] ⇨ [Pulse Contact Symbol] ⇨ [Falling Pulse Branch]	
⇨ [Ladder Symbol] ⇨ [Pulse Contact Symbol] ⇨ [Rising Pulse Close]	
⇨ [Ladder Symbol] ⇨ [Pulse Contact Symbol] ⇨ [Falling Pulse Close]	
⇨ [Ladder Symbol] ⇨ [Pulse Contact Symbol] ⇨ [Rising Pulse Close Branch]	
⇨ [Ladder Symbol] ⇨ [Pulse Contact Symbol] ⇨ [Falling Pulse Close Branch]	
⇨ [Ladder Symbol] ⇨ [Invert Operation Results]	—
⇨ [Ladder Symbol] ⇨ [Operation Result Rising Pulse]	
⇨ [Ladder Symbol] ⇨ [Operation Result Falling Pulse]	
⇨ [Inline Structured Text] ⇨ [Insert Inline Structured Text Box]	Page 134 Inserting inline structured text
⇨ [Inline Structured Text] ⇨ [Display Template]	Page 149 Displaying syntax templates
⇨ [Inline Structured Text] ⇨ [Mark Template (Left)]	
⇨ [Inline Structured Text] ⇨ [Mark Template (Right)]	
⇨ [Inline Structured Text] ⇨ [Register Label]	Page 149 Registering undefined labels
⇨ [Inline Structured Text] ⇨ [Comment Out of Selected Range]	Page 149 Batch comment out/uncomment of a program
⇨ [Inline Structured Text] ⇨ [Disable Comment Out of Selected Range]	
⇨ [Edit FB Instance]	Page 130 Editing FB instance names
⇨ [Change FB/FUN Data]	Page 130 Replacing FB instances Page 133 Replacing functions
⇨ [I/O Argument] ⇨ [Increment Argument]	Page 133 Adding/deleting arguments
⇨ [I/O Argument] ⇨ [Delete Argument]	
⇨ [Documentation] ⇨ [Edit Device/Label Comment]	Page 135 Entering/editing comments
⇨ [Documentation] ⇨ [Edit Statement]	Page 136 Entering/editing statements
⇨ [Documentation] ⇨ [Edit Note]	Page 138 Entering/editing notes
⇨ [Documentation] ⇨ [Delete Device/Label Comment]	—
⇨ [Documentation] ⇨ [Statement/Note Batch Edit]	Page 136 Entering/editing statements Page 138 Entering/editing notes
⇨ [Documentation] ⇨ [Show/Hide of Navigation Window]	Page 137 Line statement to be displayed in the tree
⇨ [Easy Edit] ⇨ [Connect Horizontal Line to Right-Side Ladder Symbol]	—
⇨ [Easy Edit] ⇨ [Connect Horizontal Line to Left-Side Ladder Symbol]	
⇨ [Easy Edit] ⇨ [Enter/Delete Horizontal Line to Rightward]	
⇨ [Easy Edit] ⇨ [Enter/Delete Horizontal Line to Leftward]	
⇨ [Easy Edit] ⇨ [Enter/Delete Vertical Line to Downward]	
⇨ [Easy Edit] ⇨ [Enter/Delete Vertical Line to Upward]	
⇨ [Easy Edit] ⇨ [Switch Ladder Symbol Invert]	Page 128 Switching methods for contacts/instructions
⇨ [Easy Edit] ⇨ [Switch Pulse/Switch SET and RST Instructions]	
⇨ [Easy Edit] ⇨ [Change Statement/Note Type]	—
⇨ [Easy Edit] ⇨ [Instruction Partial Edit]	
⇨ [Temporarily Change Ladders] ⇨ [Temporarily Change Ladders]	Page 141 Temporary changing
⇨ [Temporarily Change Ladders] ⇨ [Restore the Changes]	Page 142 Applying/restoring the changes
⇨ [Temporarily Change Ladders] ⇨ [Apply the Changes]	

[Find/Replace]	
⇨ [Line Statement List]	Page 137 Displaying list
⇨ [Jump]	Page 143 Jump
⇨ [Jump to Next Ladder Block Start]	—
⇨ [Jump to Previous Ladder Block Start]	

[View]	
⇨ [Toolbar] ⇨ [Ladder]	—
⇨ [Comment Display]	Page 135 Entering/editing comments
⇨ [Statement Display]	Page 136 Entering/editing statements
⇨ [Note Display]	Page 138 Entering/editing notes
⇨ [Display Lines of Monitored Current Value]	—
⇨ [Display Format for Device Comment]	
⇨ [Change Display Format of Device/Label Name] ⇨ [1 Cell Display]	Page 123 Configuration of ladder editor
⇨ [Change Display Format of Device/Label Name] ⇨ [Wrapping Ladder Display]	
⇨ [Outline] ⇨ [Expand/Collapse of Outlines]	—
⇨ [Outline] ⇨ [Expand/Collapse of All Outlines]	
⇨ [Outline] ⇨ [Show/Hide of Outlines]	Page 123 Configuration of ladder editor
⇨ [Display Device]	
⇨ [Text Size] ⇨ [Bigger]	—
⇨ [Text Size] ⇨ [Smaller]	
⇨ [Open Label Setting of Selected Element] ⇨ [Open in Front]	
⇨ [Open Label Setting of Selected Element] ⇨ [Tile Horizontally]	
⇨ [Open Program Body of Selected Element] ⇨ [Open in Front]	
⇨ [Open Program Body of Selected Element] ⇨ [Tile Horizontally]	
⇨ [Open Label Setting] ⇨ [Open in Front]	
⇨ [Open Label Setting] ⇨ [Tile Horizontally]	
⇨ [Open Zoom Source Block]	Page 172 Creating/displaying Zooms (action/transition)
⇨ [Instruction Help]	Page 144 Displaying instruction help

Applicable menus for ST editor

[Edit]	
⇨ [Delete]	—
⇨ [Comment Out of the Selected Range]	Page 149 Batch comment out/uncomment of a program
⇨ [Disable Comment Out of Selected Range]	
⇨ [Register Label]	Page 149 Registering undefined labels
⇨ [Display Template]	Page 149 Displaying syntax templates
⇨ [Mark Template (Left)]	
⇨ [Mark Template (Right)]	

[Find/Replace]	
⇨ [Jump]	Page 150 Jump

[View]	
⇨ [Toolbar] ⇨ [ST]	—
⇨ [Open Label Setting of Selected Element] ⇨ [Open in Front]	
⇨ [Open Label Setting of Selected Element] ⇨ [Tile Horizontally]	
⇨ [Open Program of Selected Element] ⇨ [Open in Front]	
⇨ [Open Program of Selected Element] ⇨ [Tile Horizontally]	
⇨ [Open Label Setting] ⇨ [Open in Front]	
⇨ [Open Label Setting] ⇨ [Tile Horizontally]	
⇨ [Open Zoom Source Block]	Page 172 Creating/displaying Zooms (action/transition)
⇨ [Outline] ⇨ [Expand/Collapse of Outlines]	—
⇨ [Outline] ⇨ [Expand/Collapse of All Outlines]	
⇨ [Outline] ⇨ [Show/Hide of Outlines]	Page 145 Configuration of ST editor

Applicable menus for FBD/LD editor

[Edit]	
⇨ [Delete]	—
⇨ [Select Network]	Page 155 Common operations of elements
⇨ [Insert Row]	Page 156 Inserting a row
⇨ [Delete Row]	Page 156 Deleting a row
⇨ [Insert Column of Network]	Page 156 Inserting/deleting a column
⇨ [Delete Column of Network]	
⇨ [Add Element] ⇨ [Open Contact]	Page 154 Inserting from the menu/toolbar
⇨ [Add Element] ⇨ [Close Contact]	
⇨ [Add Element] ⇨ [Open Branch]	
⇨ [Add Element] ⇨ [Close Branch]	
⇨ [Add Element] ⇨ [Coil]	
⇨ [Add Element] ⇨ [Left Power Rail]	
⇨ [Add Element] ⇨ [Pulse Contact Symbol] ⇨ [Rising Pulse]	
⇨ [Add Element] ⇨ [Pulse Contact Symbol] ⇨ [Falling Pulse]	
⇨ [Add Element] ⇨ [Pulse Contact Symbol] ⇨ [Rising Pulse Branch]	
⇨ [Add Element] ⇨ [Pulse Contact Symbol] ⇨ [Falling Pulse Branch]	
⇨ [Add Element] ⇨ [Pulse Contact Symbol] ⇨ [Rising Pulse Close]	
⇨ [Add Element] ⇨ [Pulse Contact Symbol] ⇨ [Falling Pulse Close]	
⇨ [Add Element] ⇨ [Pulse Contact Symbol] ⇨ [Rising Pulse Close Branch]	
⇨ [Add Element] ⇨ [Pulse Contact Symbol] ⇨ [Falling Pulse Close Branch]	
⇨ [Add Element] ⇨ [Open Contact and Coil]	
⇨ [Add Element] ⇨ [Variable]	
⇨ [Add Element] ⇨ [Connector]	
⇨ [Add Element] ⇨ [Jump]	
⇨ [Add Element] ⇨ [Jump Label]	
⇨ [Add Element] ⇨ [Return]	
⇨ [Add Element] ⇨ [Comment]	
⇨ [Change Name]	—
⇨ [Change FB/FUN Data]	Page 155 Replacing function elements/function block elements
⇨ [Update FB/FUN]	Page 156 Updating undefined FBs/FUNs
⇨ [I/O Argument] ⇨ [Increment Argument]	Page 155 Adding/deleting arguments
⇨ [I/O Argument] ⇨ [Delete Argument]	
⇨ [Easy Edit] ⇨ [Invert Contact (Open/Close)]	Page 154 Switching methods for contacts/instructions
⇨ [Easy Edit] ⇨ [Switch Pulse]	
⇨ [Easy Edit] ⇨ [Switch SET and RST]	

[View]	
⇨ [Toolbar] ⇨ [FBD/LD]	—
⇨ [Comment Display]	Page 151 Configuration of FBD/LD editor
⇨ [Grid Display]	
⇨ [Display Execution Order]	
⇨ [Open Label Setting of Selected Element] ⇨ [Open in Front]	—
⇨ [Open Label Setting of Selected Element] ⇨ [Title Horizontally]	
⇨ [Open Program Body of Selected Element] ⇨ [Open in Front]	
⇨ [Open Program Body of Selected Element] ⇨ [Title Horizontally]	
⇨ [Open Label Setting] ⇨ [Open in Front]	
⇨ [Open Label Setting] ⇨ [Title Horizontally]	
⇨ [Open Zoom Source Block]	Page 172 Creating/displaying Zooms (action/transition)

Applicable menus for SFC editor (SFC diagram)

[Edit]	
⇨ [Delete]	—
⇨ [Select Network]	Page 166 Common operations of SFC elements
⇨ [Modify] ⇨ [Name]	Page 163 Changing a step name/step.No./step attribute/step attribute target Page 164 Changing a transition name/Transition No.
⇨ [Modify] ⇨ [Direct Expression for Transition]	Page 164 Creating a transition
⇨ [Modify] ⇨ [Qualifier]	—
⇨ [Modify] ⇨ [End Step/Jump]	Page 159 Configuration of SFC diagram editor
⇨ [Modify] ⇨ [Step Attribute]	Page 163 Changing a step name/step.No./step attribute/step attribute target
⇨ [Modify] ⇨ [No Step Attribute]	
⇨ [Modify] ⇨ [SC: Coil HOLD Step]	
⇨ [Modify] ⇨ [SE: Operation HOLD Step (without Transition Check]	
⇨ [Modify] ⇨ [ST: Operation HOLD Step (with Transition Check]	
⇨ [Modify] ⇨ [R: Reset Step]	
⇨ [Modify] ⇨ [BC: Block Start Step (with END Check)]	
⇨ [Modify] ⇨ [BS: Block Start Step (without END Check)]	
⇨ [Modify] ⇨ [Step Attribute Target]	
⇨ [Modify] ⇨ [Device]	Page 163 Changing a step name/step.No./step attribute/step attribute target Page 164 Changing a transition name/Transition No.
⇨ [Modify] ⇨ [Switch between Jump Symbol and Connection Line]	Page 165 Switching a jump/connection line
⇨ [Insert] ⇨ [Step]	Page 163 Inserting a step
⇨ [Insert] ⇨ [Transition]	Page 164 Inserting a transition
⇨ [Insert] ⇨ [Action]	Page 165 Inserting an action
⇨ [Insert] ⇨ [Jump]	Page 165 Inserting a jump
⇨ [Insert] ⇨ [Selection Branch]	Page 166 Inserting a branch below a step/transition
⇨ [Insert] ⇨ [Simultaneous Branch]	
⇨ [Insert] ⇨ [Selection Branch Leg]	Page 166 Adding a selection branch on the right side of a transition/selection condition
⇨ [Insert] ⇨ [Simultaneous Branch Leg]	Page 166 Adding a simultaneous branch on the right side of a step/simultaneous branch
⇨ [Insert FBD/LD Element] ⇨ [Left Power Rail]	—
⇨ [Insert FBD/LD Element] ⇨ [Open Contact]	
⇨ [Insert FBD/LD Element] ⇨ [Close Contact]	
⇨ [Insert FBD/LD Element] ⇨ [Open Branch]	
⇨ [Insert FBD/LD Element] ⇨ [Close Branch]	
⇨ [Insert FBD/LD Element] ⇨ [Pulse Contact Symbol] ⇨ [Rising Pulse]	
⇨ [Insert FBD/LD Element] ⇨ [Pulse Contact Symbol] ⇨ [Falling Pulse]	
⇨ [Insert FBD/LD Element] ⇨ [Pulse Contact Symbol] ⇨ [Rising Pulse Branch]	
⇨ [Insert FBD/LD Element] ⇨ [Pulse Contact Symbol] ⇨ [Falling Pulse Branch]	
⇨ [Insert FBD/LD Element] ⇨ [Pulse Contact Symbol] ⇨ [Rising Pulse Close]	
⇨ [Insert FBD/LD Element] ⇨ [Pulse Contact Symbol] ⇨ [Falling Pulse Close]	
⇨ [Insert FBD/LD Element] ⇨ [Pulse Contact Symbol] ⇨ [Rising Pulse Close Branch]	
⇨ [Insert FBD/LD Element] ⇨ [Pulse Contact Symbol] ⇨ [Falling Pulse Close Branch]	
⇨ [Insert FBD/LD Element] ⇨ [Variable]	
⇨ [Insert FBD/LD Element] ⇨ [Connector]	
⇨ [Insert FBD/LD Element] ⇨ [Comment]	
⇨ [I/O Argument] ⇨ [Increment Argument]	Page 155 Adding/deleting arguments
⇨ [I/O Argument] ⇨ [Delete Argument]	
⇨ [Easy Edit] ⇨ [Invert Contact (Open/Close)]	Page 154 Switching methods for contacts/instructions
⇨ [Easy Edit] ⇨ [Switch Pulse]	
⇨ [Edit Step/Transition]	Page 172 Editing Step No./Transition No.

[Edit]	
⇨ [Properties]	—
[Convert]	
⇨ [Convert Block]	Page 171 Converting blocks
[View]	
⇨ [Toolbar] ⇨ [SFC]	—
⇨ [Comment Display]	Page 159 Configuration of SFC diagram editor
⇨ [Display Step/Transition]	
⇨ [Grid Display]	
⇨ [Open SFC Block List]	Page 173 Displaying an SFC block list
⇨ [Open Zoom List]	Page 172 Displaying a Zoom list
⇨ [Open Label Setting of Selected Element] ⇨ [Open in Front]	—
⇨ [Open Label Setting of Selected Element] ⇨ [Tile Horizontally]	
⇨ [Open Program Body of Selected Element] ⇨ [Open in Front]	
⇨ [Open Program Body of Selected Element] ⇨ [Tile Horizontally]	
⇨ [Open Label Setting] ⇨ [Open in Front]	
⇨ [Open Label Setting] ⇨ [Tile Horizontally]	
⇨ [Open Zoom/Start Destination Block]	Page 159 Configuration of SFC diagram editor
⇨ [Open Start Source Block]	Page 163 Creating a step that makes another block activated
[Debug]	
⇨ [Control SFC Step] ⇨ [Activate the Selected Step]	Page 273 Changing active status
⇨ [Control SFC Step] ⇨ [Deactivate the Selected Step]	
⇨ [Control SFC Step] ⇨ [Activate the Selected Step Only]	

Applicable menus for SFC editor (block list)

[Edit]	
⇨ [Delete]	—
[Find/Replace]	
⇨ [Jump]	Page 174 Jump
⇨ [Block Information Find Device]	Page 174 Searching for block information
[View]	
⇨ [SFC Block List Comment]	Page 173 Displaying an SFC block list
⇨ [Device Display]	
⇨ [Open SFC Body]	Page 174 Displaying SFC diagrams
⇨ [Open Label Setting]	Page 174 Displaying local label editor

Applicable menus for label editors (global label, local label, task, structure)

[Edit]	
⇨ [Delete]	—
⇨ [Select All]	
⇨ [New Declaration (Before)]	Page 108 Editing rows
⇨ [New Declaration (After)]	
⇨ [Delete Row]	
⇨ [Import File]	Page 114 Importing/exporting files
⇨ [Export to File]	
⇨ [System Label] ⇨ [Reservation to Register System Label]	Page 117 Registering labels in system label database
⇨ [System Label] ⇨ [Reservation to Release System Label]	Page 117 Releasing system labels
⇨ [System Label] ⇨ [Import System Label]	Page 117 Importing system labels in system label database to GX Works3
⇨ [System Label] ⇨ [Reflect to System Label Database]	Page 117 Registering labels in system label database

[Edit]	
⇨ [System Label] ⇨ [Check the changes of the System Label Database]	Page 118 Importing the changes of system label database
⇨ [System Label] ⇨ [Execute Verification Synchronous with System Label]	Page 118 Verifying system label information
⇨ [Copy Device Comment]	Page 113 Copying device comments
⇨ [Delete Blank Rows]	Page 108 Moving a row upward
[View]	
⇨ [Toolbar] ⇨ [Label]	—
⇨ [Show/Hide of Label Item]	
⇨ [Display Program Editor]	

Applicable menus for device memory editor

[Edit]	
⇨ [Delete]	—
⇨ [Enter Character String]	Page 195 Setting character strings
⇨ [Clear All (All Devices)]	Page 195 Clearing whole memory of device memory
⇨ [Clear All (Displayed Devices)]	
⇨ [FILL]	Page 194 Setting values in batch
⇨ [Register/Import Device Initial Value]	Page 196 Interaction with device initial value
[View]	
⇨ [Display Format Detailed Setting]	—

Applicable menus for device comment editor

[Edit]	
⇨ [Delete]	—
⇨ [Select All]	
⇨ [Detect the Mismatched Comment]	Page 179 Detecting devices with empty cell
⇨ [Read from Sample Comment]	Page 182 Reading sample comments
⇨ [Delete Unused Device Comment]	Page 180 Deleting unused device comments
⇨ [Clear All (All Devices)]	Page 180 Clearing all device comments
⇨ [Clear All (Displayed Devices)]	
⇨ [Import File]	Page 181 Importing to/exporting from files
⇨ [Export to File]	
⇨ [Hide All Bit Specification Information]	—
⇨ [Show All Bit Specification Information]	
⇨ [Cut the Range Including Hidden Bit Specification Information]	Page 179 Creating device comments
⇨ [Copy the Range Including Hidden Bit Specification Information]	
⇨ [Paste the Range Including Hidden Bit Specification Information]	

Applicable menus for Verify Result screen

[Edit]	
⇨ [Export to File]	Page 79 Exporting data to file
[Find/Replace]	
⇨ [Next Unmatched]	Page 78 Checking verification result
⇨ [Previous Unmatched]	
[View]	
⇨ [Return to Result List]	—
⇨ [Close Detailed Result]	
⇨ [Close All Detailed Result]	

Applicable menus for Device/Buffer Memory Batch Monitor screen

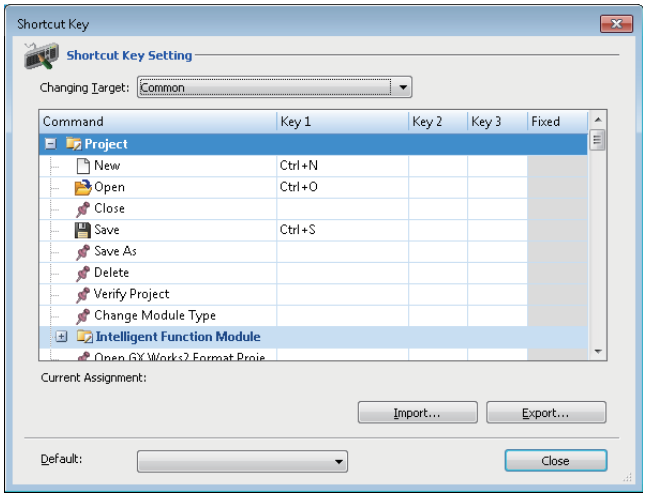
[View]	
⇨ [Display Format Detailed Setting]	—

2.5 Checking and Changing Shortcut Keys

The shortcut keys of each function can be checked and changed on the "Shortcut Key" screen.
Up to three shortcut keys can be assigned to one command.

Window

[Tool] ⇨ [Shortcut Key]



Operating procedure

1. Double-click the command cell that will change the shortcut key to edit.
2. Press the key to be assigned on the keyboard.
3. Click the [Close] button.

■Change to default setting

The shortcut key assignment can be changed to the default format by selecting the format to be set from the pull-down list in "Default".

The available formats are as follows:

- Change to GX Works3 Format: Change to the default setting.
- Change to GPPA Format: Batch change the shortcut key setting of all command keys to the same setting of GPPA
- Change to GPPW Format: Batch change the shortcut key setting of all command keys to the same setting of GX Developer.
- Change to MEDOC Format: Batch change the shortcut key setting of all command keys to the same setting of MELSEC MEDOC.

Point

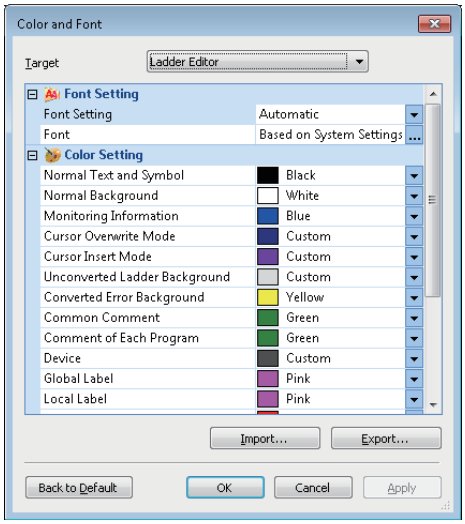
By importing the exported file (*. gks), the settings can be utilized in other personal computers.
The setting files exported from GX Works2 can be imported.

2.6 Checking and Changing Colors and Fonts

The colors and fonts used in each editor can be checked and changed on the "Color and Font" screen.
The changed color and font settings are saved in a logon user unit.

Window

[View] ⇨ [Color and Font]



Operating procedure

1. Select "User Setting" from the pull-down list in "Font Setting".
2. Set the items on the "Font" screen, and click the [OK] button.
3. Select the color from "Color Setting" to be changed, and click the [Apply] button.
4. Click the [OK] button.

Point

By importing the exported file (*. gcs), the settings can be utilized in other personal computers.
The setting files exported from GX Works2 can be imported. However, the name of the color may differ.

Precautions

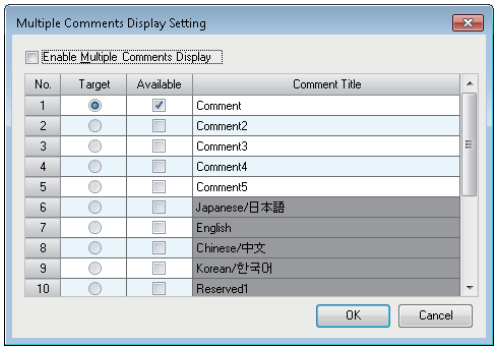
Some fonts may be displayed as garbled characters.
Change the setting to another font if this happens.

2.7 Setting Comment Display

Create multiple comments and select the comments to be displayed on each editor and each monitor screen on the "Multiple Comments Display Setting" screen.
Only five comment titles (number from 1 to 5) can be set.
The comment number from 6 to 10 are used for entering comments of predefined language.
Do not use the comment number form 11 to 16 (Reserved1 to Reserved6).

Window

[View] ⇨ [Multiple Comment Display Setting]



Operating procedure

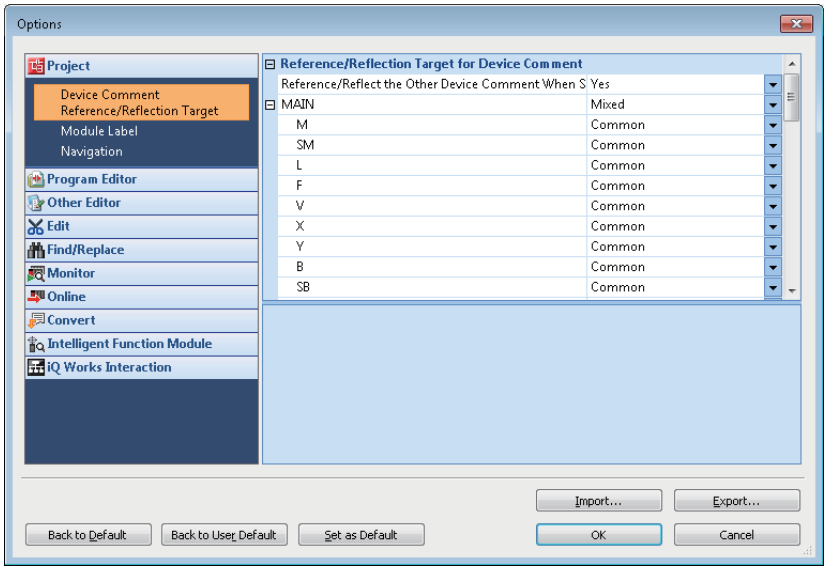
1. Select "Enable Multiple Comments Display".
2. Select "Available", and enter a comment title.
3. Select the comment to be displayed on the program editor or each monitor screen in "Target", click the [OK] button.
4. Enter comments in each row on the device comment editor.

2.8 Setting Options

Some functions and editors have an option setting function. Changing option settings enables the change of a screen display format and the settings of detailed operation for each function.

Window

[Tool] ⇨ [Options]



Operating procedure

Set the items and click the [OK] button.

Point

By importing the exported file (*. gos), the settings can be utilized in other personal computers.
The setting files exported from GX Works2 can be imported.
The setting items set to "Project" ⇨ "Device Comment Reference/Reflection Target" are not exported.

Precautions

After changing the following option setting, performing "Rebuild All" is required.

• [Tool] ⇨ [Options] ⇨ "Other Editor" ⇨ "Label Editor Common" ⇨ "Data Type Setting"

As a precautionary measure, perform the following operation before switching the CPU module to RUN.

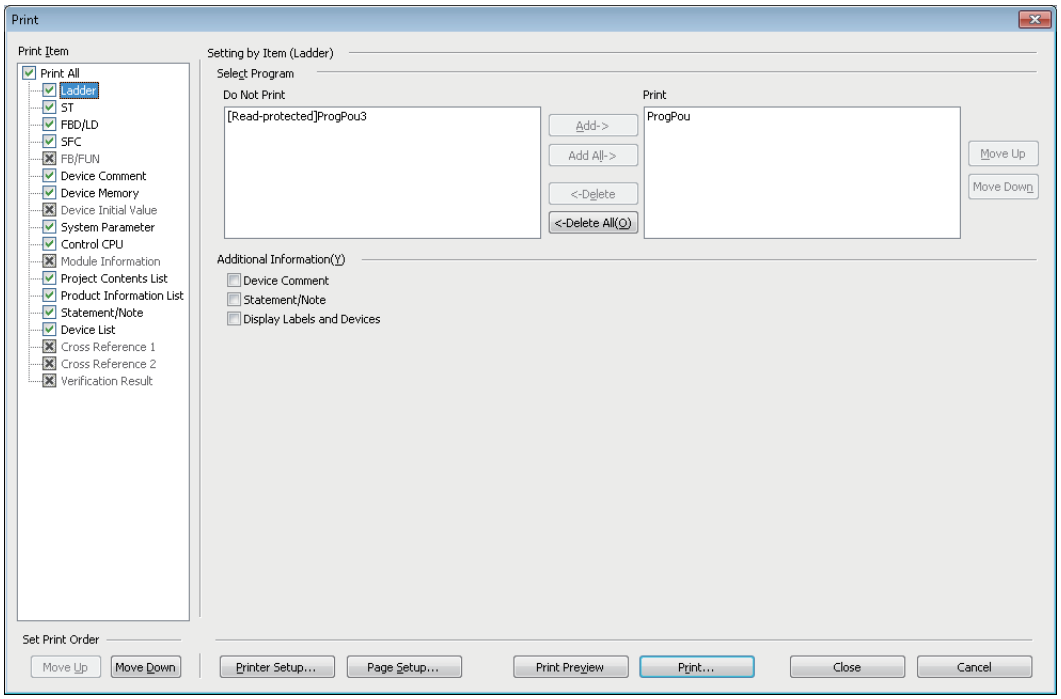
1. Reset the CPU module.
2. Clear values of devices/labels to '0' (Including latch).
3. Clear values of file registers to '0'.

2.9 Printing Data

This section explains how to print the data created in GX Works3.

Window

[Project] ⇨ [Print] ( )



When no data exist or only read-protected data exist in "Print" and "Do Not Print" list, × is displayed on the checkbox of the Print Item.

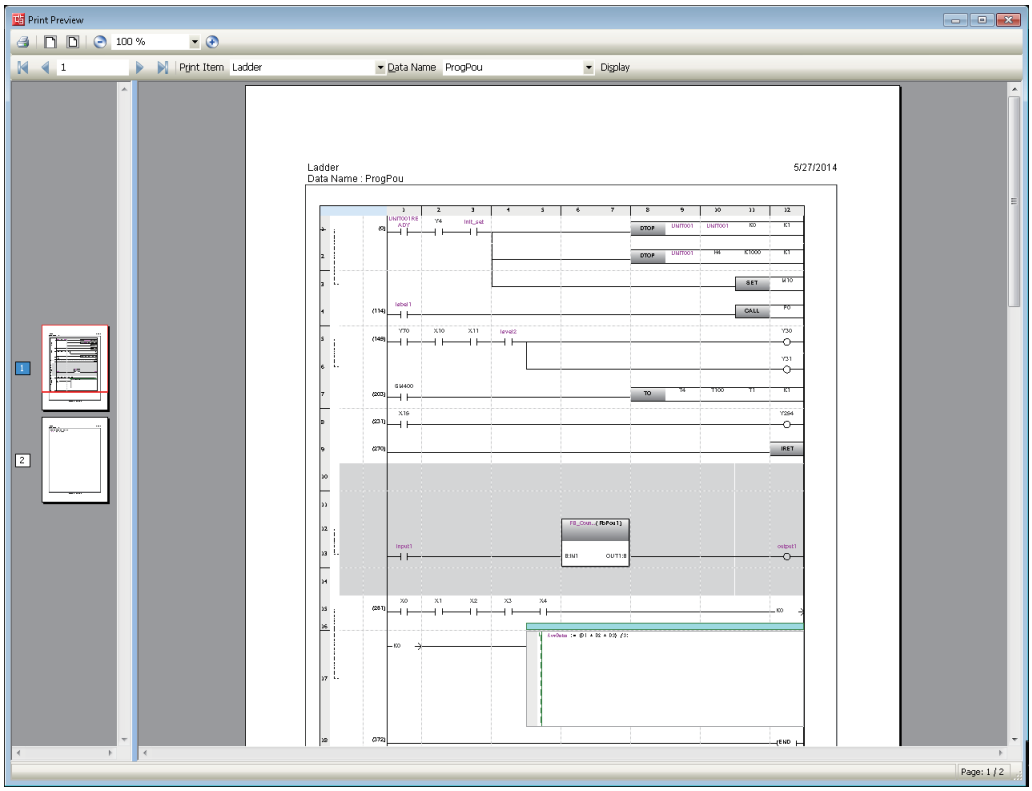
Operating procedure

1. Select the data to be printed, and configure each setting.
2. Set the printer and page by clicking the [Printer Setup] button and [Page Setup] button.
3. Click the [Print] button.

Previewing print images

Check the print images of each data.

Window



Operating procedure

1. Select [Project] ⇨ [Print Preview].
2. Click the [Print Preview] button on the "Print" screen.
3. When switching the data to be previewed, select "Print Item" and "Data Name", and click the [Display] button.

Considerations

■Display of print previews and printing data

Displaying print previews and printing data cannot be performed in the following situations.

- The print target data do not exist
- The print target screen is being monitored (Start printing after stopping the monitoring.)

■Printing large volume data

Large volume data may not be printed at all or printed half way due to the limitations of printer driver or Windows® print spooler. In this case, print the data by any of the following methods.

- Split the data by setting a print range
- Select "Print directly to the printer" in the [Advanced] tab of the property screen (select [Control Panel] ⇨ [Devices and Printers] from Windows® Start*1)
- Set "Output by Item" for the print job output on the "Printer Setup" screen.

*1 Select [All apps] on the Start screen or [Start] ⇨ [All Programs].

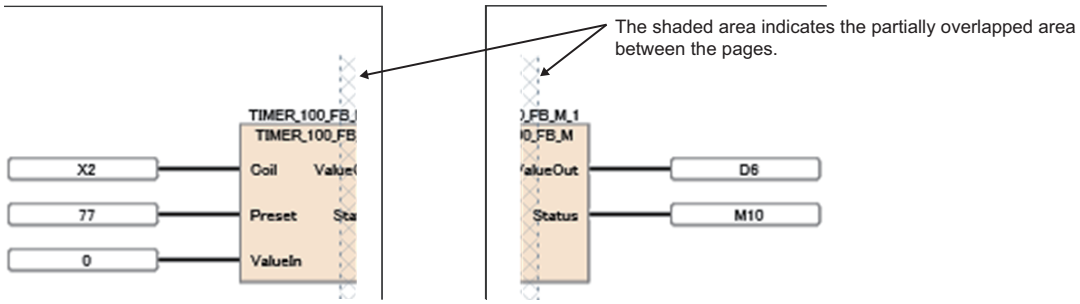
■Outputting the print contents on a file

- When the print content is output to a file, the "Save As" screen may be displayed in the background of other screen. Press the Alt + Tab keys or Alt + Esc keys to bring the screen in the foreground.
- GX Works3 does not respond when the GX Works3 operation is performed while the "Save As" screen is displayed. When the message is displayed, select "Wait for the program to respond". Selecting "Close the program" results in loss of unsaved data.

■Printing FBD/LD programs

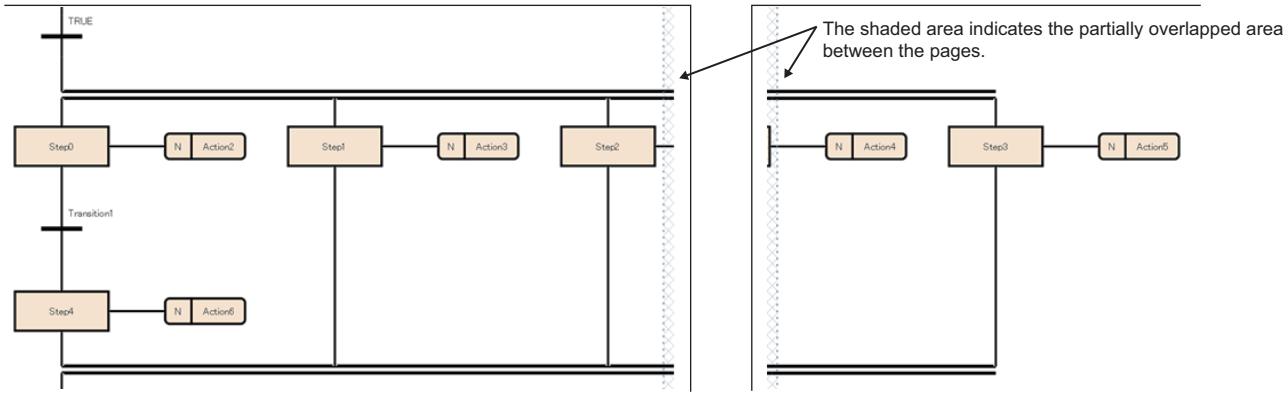
Since the elements can be placed at will on the FBD/LD editor, it may be divided into several pages and printed out as shown below.

Check the layout in the print preview before printing.



■Printing SFC programs

- Statements and notes in a Zoom are excluded from the print target of Statement/Note.
- For the SFC diagram editor, SFC elements may be divided into several pages and printed out as shown below. Check the layout in the print preview before printing.



■Printing the product information list

The information in the specified CSV file is printed out within the range of 500 rows × 20 columns.

MEMO

PART 2 SYSTEM DESIGN AND SETTINGS

This part explains the system design such as project management and parameter/label settings.

3 PROJECT MANAGEMENT

4 CREATING MODULE CONFIGURATION DIAGRAM AND SETTING PARAMETERS

5 REGISTERING LABELS

3 PROJECT MANAGEMENT

This chapter explains the basic operations and the project management.

3.1 Project File and Data Configuration

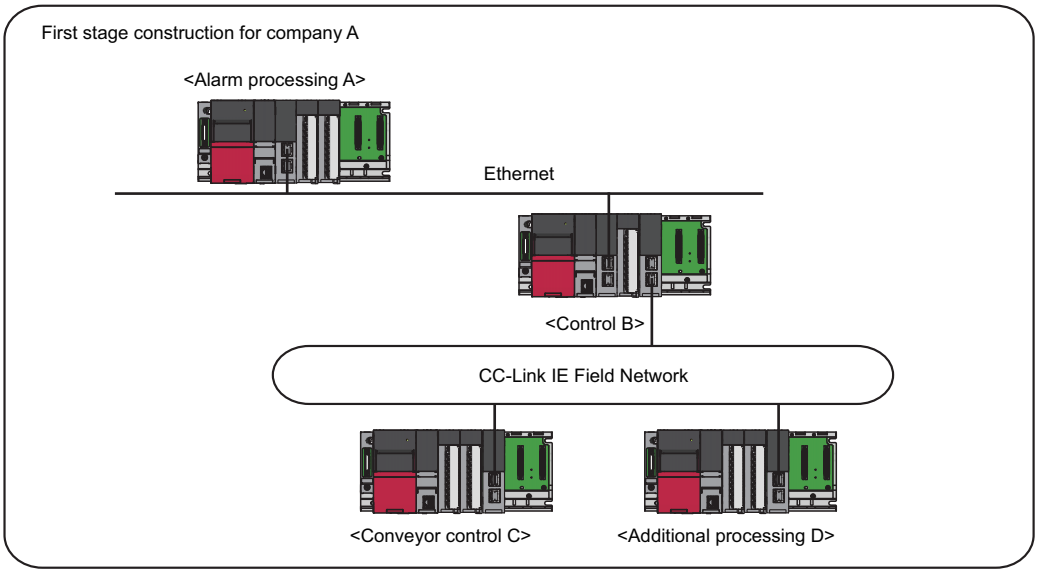
The projects created in GX Works3 are saved in workspace formats or single file formats.
The data created in the project is displayed on the Project view.

File format

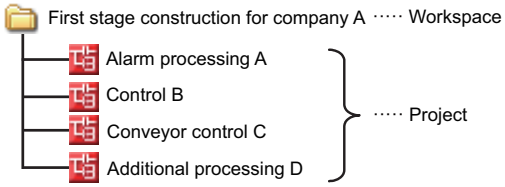
Workspace format

A workspace manages multiple projects at once.
When configuring a system composed with multiple CPU modules, a project file needs to be created for each CPU module.
Multiple project files in a same system can be managed by saving in workspace format.

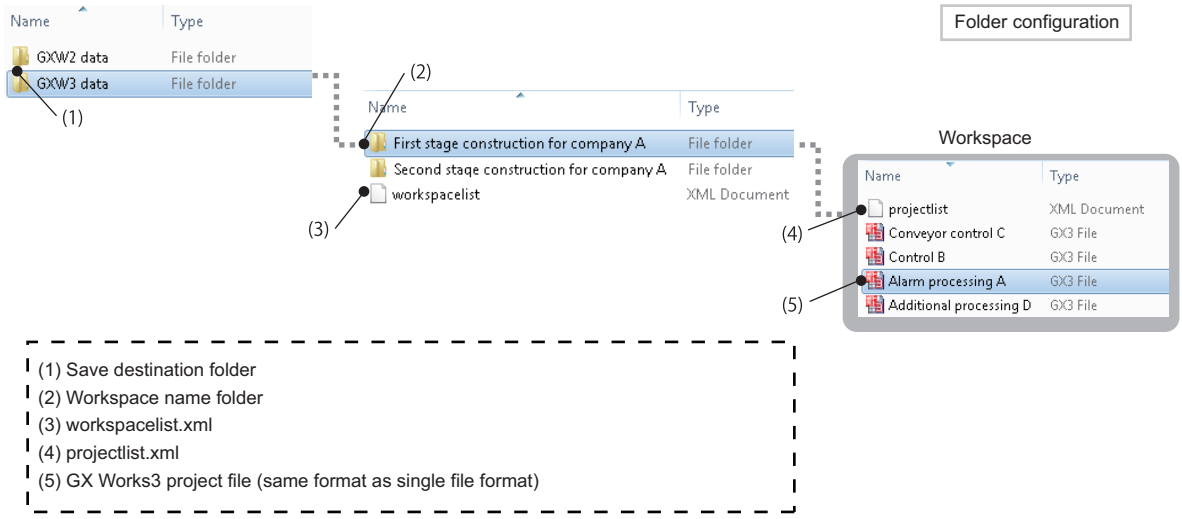
<System configuration example>



<GX Works3 project management>



■Configuration of workspace/project



(1) Save destination folder
(2) Workspace name folder
(3) workspacelist.xml
(4) projectlist.xml
(5) GX Works3 project file (same format as single file format)

● When editing configurations or names

Do not change or delete workspace configurations or project names by using an application such as Windows® Explorer. The projects that do not contain actual data may remain in a project list of the function such as the "Open Project" function. (Select [Project] ⇨ [Delete] to delete the project.)

● When copying a project

For copying a project using Windows® Explorer, perform either of the following operations.

The configuration of workspace/project will not be collapsed.

- Copy the entire save destination folder ((1) above).
- Copy the target workspace name folder ((2) above) and "workspacelist.xml" ((3) above).
- Copy the GX Works3 project file that has the same name.

Single file format

Single file format is a format that does not need a workspace.

Projects can be managed without being aware of the folder configuration and the file configuration by saving projects in the single file format.

Thereby, operations such as changing project names, copying and pasting projects, and sending and receiving data, can easily be performed using an application such as Windows® Explorer.

Backup of safety project

Back up safety projects by a system administrator, and save the data securely so that they can be restored when required.

Make sure to back up a project by selecting [Project] ⇨ [Save As] in GX Works3.

Data configuration

Data displayed on the Project view

The data displayed on the Project view in tree format are as shown below. (The data names are default.)

■Module configuration data

Creation method: ☞ Page 85 Creating Module Configuration Diagram

Figure	Item	Description
	Module Configuration Diagram	A data that displays target system of a project graphically.

■Program data

Creation method: ☞ Page 71 Creating Data

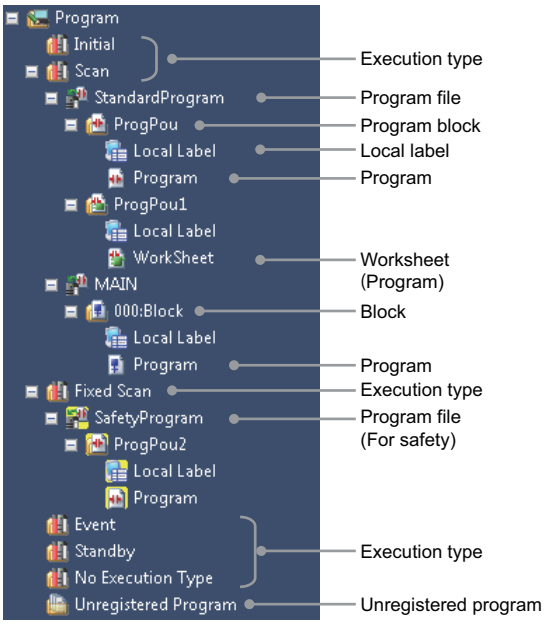
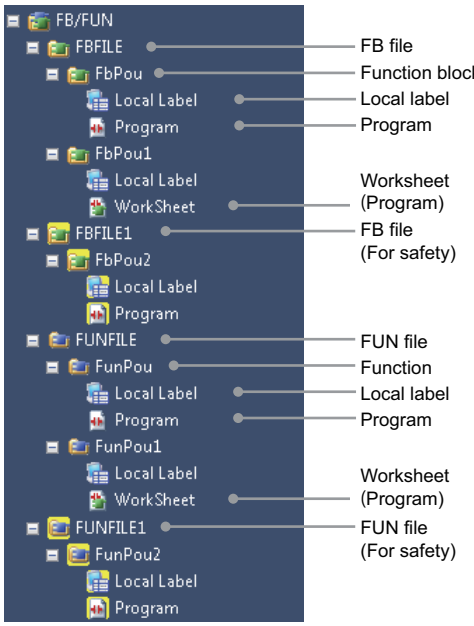
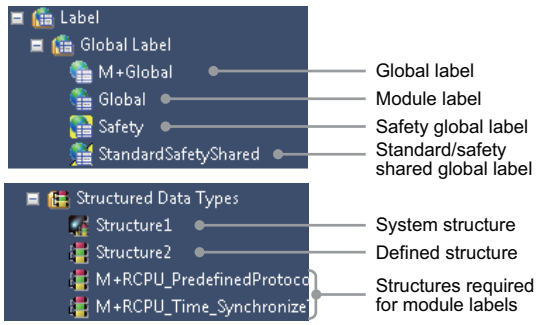
Figure	Item	Description
	Execution types	A setting for when a program operates. When executing programs in a CPU module, the programs require to be registered to any of the execution types. ☞ Page 122
	Program file	A file that manages programs. Create the file in unit of execution process. The created files are written to a CPU module in unit of program files.
	Program block (POU)	A data that composes a program.
	Local label	A label data that can be used only in the defined program block.
	Program	A program data that is created using Ladder Diagram or Structured Text.
	Worksheet (Program)	A program data that is created using Function Block Diagram/Ladder Diagram.
	Block	Data that composes an SFC block.
	Program	SFC diagram data that is created using Sequential Function Chart.
	Program file (For safety)	A file that manages a safety program.
	Unregistered program	A folder that temporarily stores program files of which execution type is undefined. The stored program files will not be executed even when writing them to a programmable controller.

Figure	Item	Description
	FB file	A file that manages function blocks. The FB files are written to a CPU module in unit of FB file. ☞ Page 214 Creating Function Blocks
	Function block (POU)	A data that configures the program of function block.
	Local label	A label data that can be used only in the defined function block.
	Program	A function block data that is created using Ladder Diagram or Structured Text. This data is referred to as 'FB program' in this manual.
	Worksheet (Program)	A function block data that is created using Function Block Diagram/Ladder Diagram.
	FB file (For safety)	A file that manages function blocks of a safety program.
	FUN file	A file that manages functions. The function files are written to a CPU module in unit of FUN file. ☞ Page 217 Creating Functions
	Function (POU)	A data that configures programs of function.
	Local label	A label data that can be used only in the defined function.
	Program	A function data that is created using Ladder Diagram or Structured Text. This data is referred to as 'FUN program' in this manual.
	Worksheet (Program)	A function data that is created using Function Block Diagram/Ladder Diagram.
	FUN file (For safety)	A file that manages functions of a safety program.

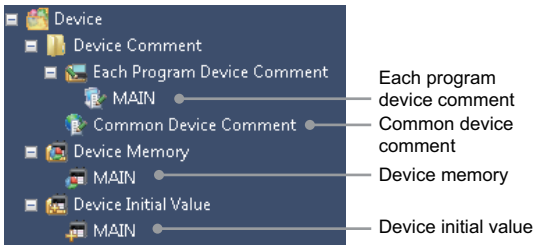
■Label data

Creation method: ☞ Page 71 Creating Data

Figure	Item	Description
	Global label	A label data that can be accessed from all the program blocks and function blocks in the project. ☞ Page 107 Registering Labels
	Module label	A label to access I/O signals or buffer memory of modules. ☞ Page 115 Registering Module Labels
	Safety global label	A global label for a safety program.
	Standard/safety shared global label	A global labels that can be used in both standard programs and safety programs.
	System structure	A structure registered in the system label database.
	Defined structure	A data that defines structure as data type. This can be used as a data type for all labels which can be defined in the project except for recursive definitions in the defined structure.
	Structures required for module labels	The structures required for module labels are registered automatically.

■Device data

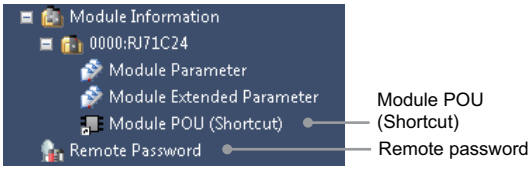
Creation method: ☞ Page 71 Creating Data

Figure	Item	Description
	Each program device comment	A device comment data that can be used in the program files which have the same name. ☞ Page 176 Registering Device Comments
	Common device comment	A device comment data that can be commonly used in multiple programs. ☞ Page 176 Registering Device Comments
	Device memory	A data that includes the values to be written to/read from devices in a CPU module. ☞ Page 191 SETTING DEVICE MEMORY
	Device initial value	A data that defines values which are set to devices when CPU module is in RUN. ☞ Page 197 SETTING DEVICE INITIAL VALUES

■Parameter data

The structure of the tree and the creation methods: ☞ Page 93 Setting Parameters

■Other data

Figure	Item	Description
	Module POU (Shortcut)	A shortcut to the module POU that has been registered to the Element Selection window.
	Remote password	By setting a password for the CPU module, accessing from the modules other than specified RJ71EN71, serial communication module, and built-in Ethernet CPU can be prohibited. ☞ Page 310 Restricting Access from Other Than Specific Communication Route

3.2 Creating Project File

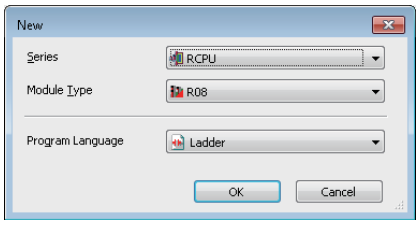
This section explains basic operations of GX Works3 such as creating, opening, and saving projects.

Creating project

For a safety project, registering a user is required when creating a new project since a user information for the User Authentication function is needed. (☞ Page 305 User management)

Window

[Project] ⇨ [New] ( )



Operating procedure

Set the items and click the [OK] button.

Precautions

When creating a safety project, set a screen saver in Windows ® to lock the personal computer automatically if the non-operated state has been continued for a certain time.
For details, refer to Windows® Help and Support.

■Unsupported series by GX Works3

When the series (QCPU (Q mode), LCPU, FXCPU) which are not supported by GX Works3 is selected, GX Works2 is started up and create a new project in it.
When GX Works2 is installed on the personal computer, it will be started up automatically.
If it is not installed, execute the 'setup.exe' in the installation DVD-ROM (Disk2) of GX Works3.

■Creating a project for RnENCPU

RnENCPU consists of the two slots that are the CPU part and the network part.
Only CPU parts is placed after creating a new data on the Module Configuration Diagram. Place a CPU extension module (_RJ71EN71) on the Module Configuration Diagram/in the "I/O Assignment Setting" of system parameter (☞ Page 86 Placing module objects), then set the network part.

Creating new project by data read from programmable controller

When the Read from PLC function is performed without creating a new project, a new project is created by data read from the CPU module and Intelligent function modules.

When a user information is registered to the project, the user authentication is required. (☞ Page 306 Logging on to CPU module)

Operating procedure

1. Start GX Works3 and select [Online] ⇨ [Read from PLC] ( ).
2. Select the series to be read on the "Series Selection" screen, and click the [OK] button.
3. Set the communication route to access the CPU module on the "Specify Connection Destination" screen, and click the [OK] button.
4. Perform the Read from PLC function on the "Online Data Operation" screen.

For the method for performing the Read from PLC function on the "Online Data Operation" screen, refer to the following section.

☞ Page 254 Reading data from programmable controller

Precautions

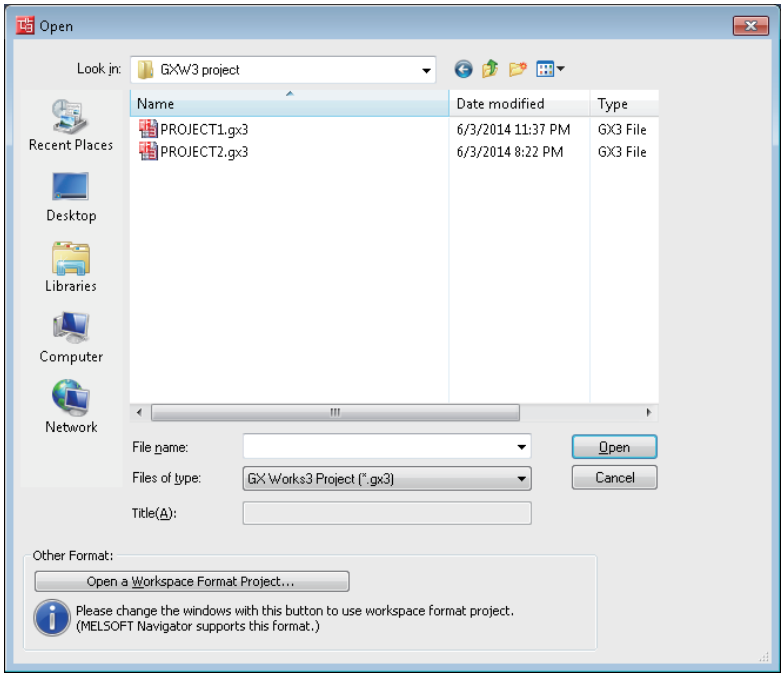
When parameters are not read from the programmable controller with data to create a new project, default parameters will be set. Check the parameter setting.

Opening project

Read the project saved on the personal computer or another data storage device.
When a user information is registered to the project, the user authentication is required. (☞ Page 306 Logging on to CPU module)

Window

[Project] ⇨ [Open] ( )



Operating procedure

Set the items and click the [Open] button.

Precautions

■Changing work space name and folder configuration

Do not change the file names and storage location of a workspace and project folders and files using an application such as Windows® Explorer.

■Opening projects being edited by other users

The project can be opened as a read-only project. However, the following functions cannot be used.

- Overwriting projects
- Changing module type

■Projects saved on a network drive or a removable media

Do not open the project directly. Open it after saving it to the personal computer or another data storage device.

■Opening a safety project

Set the screen saver in Windows® to lock the personal computer when no user activity has been sensed for a certain time. For details, refer to Windows® Help and Support.

Opening GX Works2 format project

Open the project created in GX Works2 by changing the module type with GX Works3.

This function is supported by the projects of Universal model QCPU/Universal model High-speed Type QCPU/FXCPU (FX3U/FX3UC) only.

The module type of each project will be changed as below.

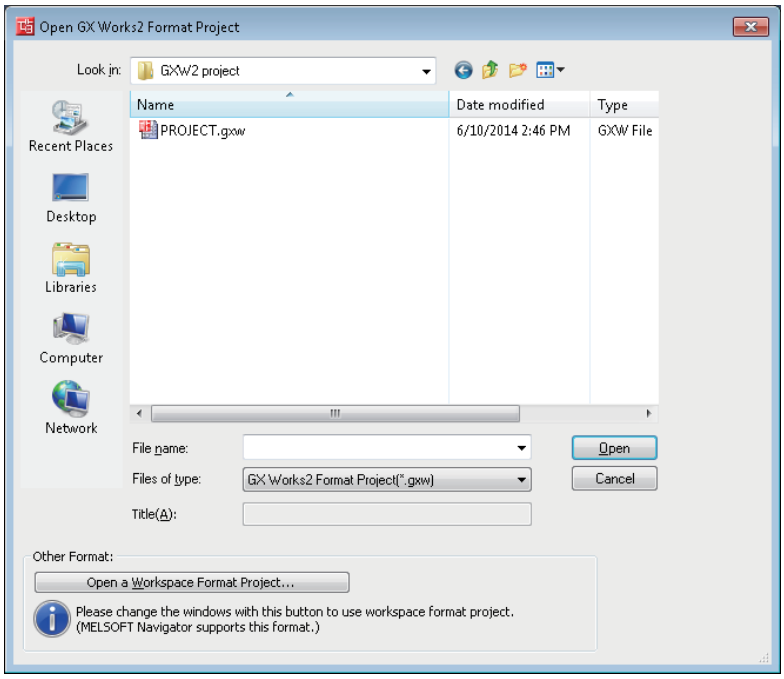
Module type before change	Module type after change
Universal model QCPU/Universal model High-speed Type QCPU	R120CPU
FXCPU (FX3U/FX3UC)	FX5UCPU

The supported version of GX Works3 differs depending on the module type. For details, refer to the following section.

☞ Page 332 Functions Added to/Changed from Previous Version

Window

[Project] ⇨ [Open Other Format File] ⇨ [Open GX Works2 Format Project] ⇨ [Open Project]



Operating procedure

1. Specify the project and click the [Open] button.
2. Read the displayed message, and click the [OK] button.

The changed content of the project data is displayed on the Output window.

Precautions

- If a program that contains a compilation error exists in GX Works2, the project cannot be opened. Check if the program can be compiled in GX Works2.
- The project to which the user management or access authority is set cannot be opened. Clear the setting in GX Works2.
- Once the module type is changed, "Label Area Capacity" and "Latch Label Area Capacity" of the CPU parameters are set to "0 K Word". Set the capacity in accordance with the label capacity of the project.

■Data that is changed at changing module type

Operation at changing module type	Setting items of GX Works2		Remarks
Changed in accordance with the target module type	• PLC Parameter	PLC Name	Check the settings after the change.
		PLC System*1,*2	
		PLC File Setting	
		PLC RAS*1	
		Boot File	
		Program*1	
		SFC*3	
		Device*1,*2	
		I/O Assignment*4	
		Multiple CPU Setting*1,*4	
		Built-in Ethernet Port Setting	
	• Network Parameter	CC-Link IE setting	
		Ethernet setting	
		CC-Link setting*2	
	• Intelligent function module parameter	QD75 Positioning module*2,*4	
	• FB/FUN • Structured Ladder/FBD program*5 • SFC program*1,*3 • Device comment for devices other than SM/SD*6		
Returns to default/ Data are deleted	• Options (other than "Device Comment Reference/Reflection Target") • System label • Device comments of SM/SD*6 • Connection destination • Remote password		Set with GX Works3.
Deleted	• PLC Parameter	Communication Head Setting	Not supported by GX Works3.
		Built-in I/O Function Setting	
		Serial Communication	
		Built-in Serial Setting	
		Adapter Serial Setting	
		Memory Capacity	
		Special Function Block	
		Positioning	
		Operation Setting	
	• Redundant Parameters	Operation mode setting	
		Tracking setting	
	• Network Parameter	MELSECNET/10 and MELSECNET/H setting	
		MELSECNET/H Remote I/O setting	
	• Programs of user library that are not registered to the program setting • Task settings • Device/Label Automatic-Assign Setting • Project Revision • Device memory (Only when the module type is changed to FX5UCPU)		

*1 Check/set with GX Works3 after the change since some items are returned to the default or their data are deleted.
*2 The data are deleted when the module type is changed to FX5UCPU.
*3 To read SFC programs, GX Works2 of which version is 1.535H or later requires to be installed in a personal computer.
*4 The items which are not supported by GX Works3 are deleted.
*5 To read Structured Ladder/FBD programs, GX Works2 of which version is 1.519R or later requires to be installed in a personal computer.
*6 The data are moved to SM8000 or higher/SD8000 or higher when the module type is changed to FX5UCPU.

■Instructions that are not supported by MELSEC iQ-R series/MELSEC iQ-F series modules

If an instruction which is not supported by MELSEC iQ-R series/MELSEC iQ-F series modules is included in the program created with GX Works2, the instruction will be changed to the program in which SM4095/SD4095 is used.

In addition, the FBs/FUNs that are not supported by the FBD/LD editor will be undefined FBs/FUNs.

Modify the program with the instruction equivalent to the one supported by MELSEC iQ-R series/MELSEC iQ-F series.

📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

📖 MELSEC iQ-F FX5 Programming Manual (Instructions, Standard Functions/Function Blocks)

■Maker-defined FUN that requires the replacement of the instruction names

The data type for a number of Maker-defined FUNs of GX Works2 is different from the one for the instructions and the arguments of GX Works3. Thereby, the data type mismatch error may occur when utilizing a program of GX Works2. In that case, replace the instruction names to the appropriate ones.

The following list shows instruction names that requires the replacement → instruction names after replacing

- BAND → BAND_U
- BANDP → BANDP_U
- BINDA → BINDA_U
- BINDAP → BINDAP_U
- BKMINUS → BKMINUS_U
- BKMINUSP → BKMINUSP_U
- BKPLUS → BKPLUS_U
- BKPLUSP → BKPLUSP_U
- DABIN → DABIN_U
- DABINP → DABINP_U
- DBAND → DBAND_U
- DBANDP → DBANDP_U
- DBINDA → DBINDA_U
- DBINDAP → DBINDAP_U
- DBKMINUS → DBKMINUS_U
- DBKMINUSP → DBKMINUSP_U
- DBKPLUS → DBKPLUS_U
- DBKPLUSP → DBKPLUSP_U
- DDABIN → DDABIN_U
- DDABINP → DDABINP_U
- DDEC → DDEC_U
- DDEC P → DDECP_U
- DDIVISION → DDIVISION_U
- DDIVISIONP → DDIVISIONP_U
- DEC → DEC_U
- DECP → DECP_U
- DGBIN → DGBIN_U
- DGBINP → DGBINP_U
- DGRY → DGRY_U
- DGRYP → DGRYP_U
- DINC → DINC_U
- DINCP → DINCP_U
- DIVISION → DIVISION_U
- DIVISIONP → DIVISIONP_U
- DLIMITP → DLIMITP_U
- DMAXP → DMAXP_U
- DMEAN → DMEAN_U
- DMEANP → DMEANP_U
- DMINP → DMINP_U
- DMINUS → DMINUS_U
- DMINUSP → DMINUSP_U
- DMULTI → DMULTI_U
- DMULTIP → DMULTIP_U
- DPLUS → DPLUS_U
- DPLUSP → DPLUSP_U
- DSCL → DSCL_U
- DSCL2 → DSCL2_U
- DSCL2P → DSCL2P_U
- DSCLP → DSCLP_U
- DSORTD → DSORTD_U
- DSTR → DSTR_U
- DSTRP → DSTRP_U
- DVAL → DVAL_U
- DVALP → DVALP_U
- DWSUM → DWSUM_U
- DWSUMP → DWSUMP_U
- DZONE → DZONE_U
- DZONEP → DZONEP_U
- GBIN → GBIN_U
- GBINP → GBINP_U
- GRY → GRY_U
- GRYP → GRYP_U
- INC → INC_U
- INCP → INCP_U
- LIMITP → LIMITP_U
- MAXP → MAXP_U
- MEAN → MEAN_U
- MEANP → MEANP_U
- MINP → MINP_U
- MINUS → MINUS_U
- MINUSP → MINUSP_U
- MULTI → MULTI_U
- MULTIP → MULTIP_U
- PLUS → PLUS_U
- PLUSP → PLUSP_U
- SCL → SCL_U
- SCL2 → SCL2_U
- SCL2P → SCL2P_U
- SCLP → SCLP_U
- SORTD → SORTD_U
- STR → STR_U
- STRP → STRP_U
- VAL → VAL_U
- VALP → VALP_U
- WSUM → WSUM_U
- WSUMP → WSUMP_U
- ZONE → ZONE_U
- ZONEP → ZONEP_U

■Devices that are not supported by MELSEC iQ-R series/MELSEC iQ-F series module

- When the devices, which are not supported by MELSEC iQ-R series/MELSEC iQ-F series module, are used in a program created with GX Works2, they are changed to SM4095/SD4095. In addition, when GX Works2 project consists of ladder programs, the devices which are changed to character string by an instruction are changed to "SM4095"/"SD4095". To search character strings, refer to the search function. (☞ Page 205 Searching/Replacing Character Strings)
- When S devices or R devices have been set to the device memory data/device initial value data created with GX Works2, they are deleted.

■Number of device points and start/end device

The setting range for the number of device points may differ between the CPU type of a project created with GX Works2 and a MELSEC iQ-R series/MELSEC iQ-F series module. When the value which has been set to the parameter cannot be set for MELSEC iQ-R series/MELSEC iQ-F series modules, it will be corrected to the value configurable by a MELSEC iQ-R series module, or it will be returned to the default value of MELSEC iQ-F series after changing the module type.

■Increase of the number of device points due to the differences in local device setting units

Although the bit devices and word devices of local devices are set in one point units for Q series, the bit devices, word devices (excluding T/ST/C), and T/ST/C for MELSEC iQ-R series are set in 64-point units, 4-point units, and 32-point units, respectively. Since the setting units are different between Q series and MELSEC iQ-R series, the CPU parameters are automatically changed after changing the module type, resulting in the increase of the number of device points.

■Device name and label name display

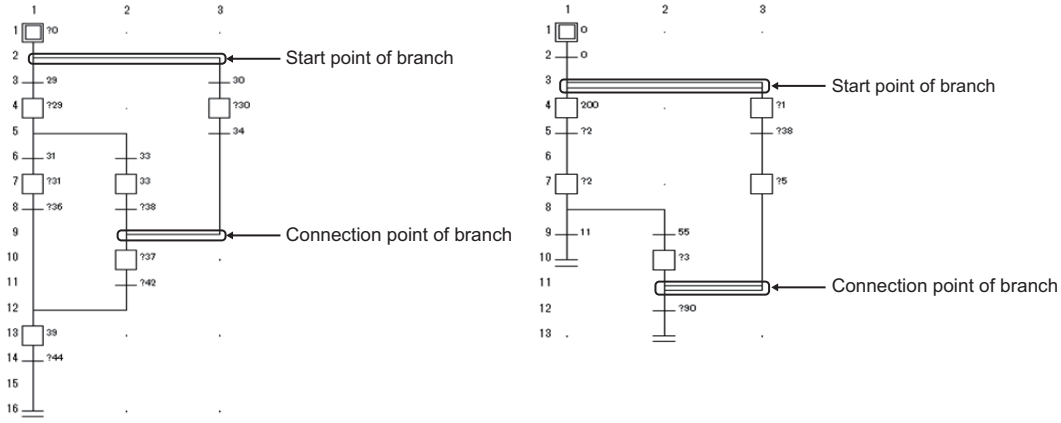
- For a device that has been set as the local device with GX Works2, '#' is added in front of the device name.
- When a label used in GX Works2 has a name starting with "M+" or "P+", the "+" of the label name is replaced with "_".

■Structured Ladder/FBD program

- The execution order is handled differently between Structured Ladder/Function Block Diagram language in GX Works2 and Function Block Diagram/Ladder Diagram language in GX Works3. Display the execution order and check it.
- The performances of return elements are different between programs written in Structured Ladder/Function Block Diagram language of GX Works2 and programs written in Function Block Diagram/Ladder Diagram language of GX Works3. For details on return elements in GX Works3, refer to the following section.
  (☞ Page 153 Common element)
- The number of I/O arguments of some function elements/function block elements is different between Structured Ladder/Function Block Diagram language in GX Works2 and Function Block Diagram/Ladder Diagram language in GX Works3. In this case, the element is displayed as undefined FB/FUN. Select [Edit] ⇨ [Update FB/FUN] to update the definition information, and correct the program.
- When a program in which members of general FB are referenced is created in Structured Ladder/Function Block Diagram of GX Works2, a conversion error may occur after opening the program with GX Works3. In this case, check the member name and class of the general FB, and correct the program.
- Inverting the contact is available for I/O arguments of function elements/function block elements in Function Block Diagram/Ladder Diagram language of GX Works3. (☞ Page 154 Switching methods for contacts/instructions)
  Connectors of those elements are deleted when the elements other than above are inverted in a Structured Ladder/FBD program of GX Works2. Modify the program to provide equivalent operations to the Structured Ladder/FBD program in GX Works2.
- Function Block Diagram/Ladder Diagram language in GX Works3 does not support wired OR of FBD elements. Correct the program to provide the equivalent operation to the Structured Ladder/FBD program of GX Works2 since a conversion error is caused.
- When a pointer branch instruction (CJ, SCJ, JMP), a jump, or a subroutine program is used, a conversion error may occur. For more details, refer to the following manual.
  (📖 MELSEC iQ-R Programming Manual (Program Design))
- The program layout may be slightly changed due to the difference between the Structured Ladder/FBD editor of GX Works2 and the FBD/LD editor of GX Works3. Correct the layout.
- Ladder block label used in the Structured Ladder/FBD of GX Works2 is defined as the pointer type in a local label.

■SFC program

- Pay attention to the data size, SFC programs in GX Works3 have a larger data size of symbolic information than the one in GX Works2.
- POUs that do not exist in the program settings of structured projects in GX Works2 cannot be read. Move the POUs to the program settings before reading.
- When label names such as "Step0" exist on an SFC diagram in GX Works2, the label names may duplicated with step names and label names in an SFC block after reading the SFC program. Change the step names/label names to avoid duplication.
- When SFC programs in MELSAP-L format have been read, they are converted into the equivalent programs in MELSAP3 format.
- SFC programs including multiple initial steps cannot be read. When multiple initial steps exist in a block, modify the program to the one that includes one initial step before reading.
- The settings to show the output only and not to show the coils in a Zoom, cannot be configured in GX Works3. Therefore, 'LD SM400' will be added to the ladder block which does not include the coils when displaying it in MELSAP3 format.
- An SFC program such as shown below cannot be read because the connection point of branch is not placed on the branch which the start position are placed on the left side. Modify the program to place the start and connection points on the left side of same branch before reading.



- When an SFC program, which includes jumps, is read, the SFC diagram is back to the initial status (that initial steps, transitions, and end steps are placed). Modify the program not to use jumps before reading.
  In another way, create an SFC diagram after reading, and connect the jumps to the read Zoom.

■Step relay/SFC block device

Step relay/SFC block device may not be available on the ST editor and the FBD/LD editor.
Modify the program to provide the equivalent operation to the ST program and Structured Ladder/FBD program of GX Works2.
For details, refer to the following manual.
📖 MELSEC iQ-R Programming Manual (Program Design)

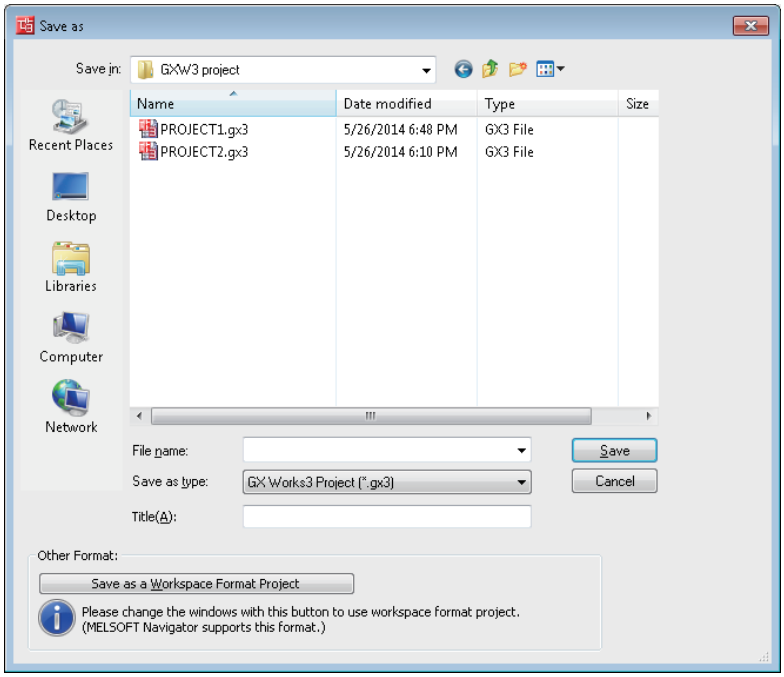
Saving project

Save the project on the personal computer or another data storage device.

Saving projects under the specified name

Window

[Project] ⇨ [Save As]



When saving the project with the workspace format, click the [Save as a Workspace Format Project] button to switch the screen.

Operating procedure

Set the items and click the [Save] button.

Overwriting projects

Operating procedure

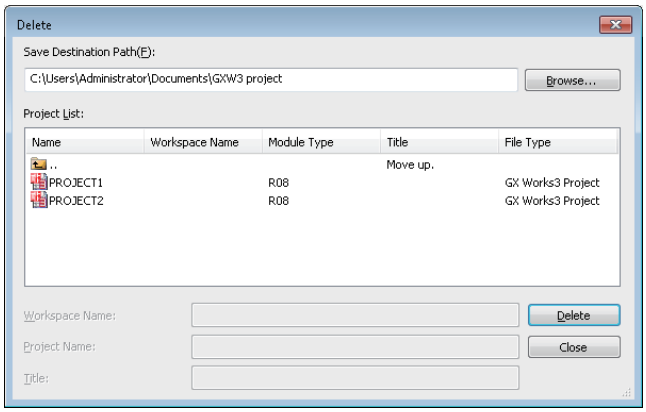
Select [Project] ⇨ [Save] (💾).

Deleting project

Delete the project saved on the personal computer or another data storage device.

Window

[Project] ⇨ [Delete]



Operating procedure

Select the project ( ) to be deleted, and click the [Delete] button.

Precautions

When the project file does not exist after deleting a workspace format project, the message to confirm whether to delete the workspace itself is displayed. If the project is deleted, the workspace folder is deleted, but workspacelist.xml remains.

3.3 Creating Data

This section explains how to create each data.

Project data

■Character strings that cannot be used

Refer to the following section.

☞ Page 334 Unusable Character String (Reserved Word)

■Maximum number of data that can be created in one project

The number of data that can be created in each data type is as follows:

Data type name	RCPU	FX5CPU
Program block	2048	2048
Function block and function	8192	960
FB file	256	15
Other data	800	800

Precautions

For data names, use the characters in the Unicode Basic Multilingual Plane.

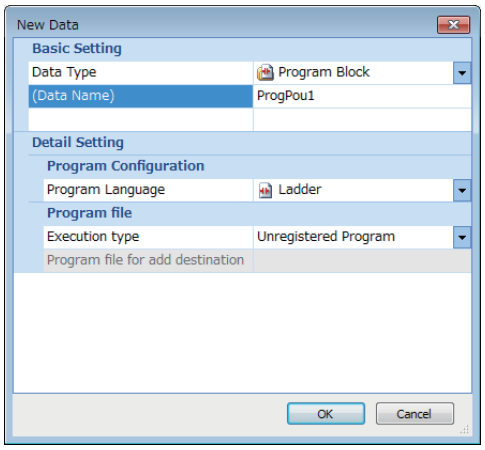
If the characters outside the Unicode Basic Multilingual Plane are specified, they may not be edited in the program.

Creating projects

Create new data in a project.

Window

Select the data on the Project view, then select [Project] ⇨ [Data Operation] ⇨ [New Data] ( )/right-click and select [New Data] ( ) from the shortcut menu.



The category column of data to select "Standard"/"Safety"/"Standard/safety shared" is displayed for a safety project. For the details on the data to select, refer to the following section.

☞ Page 19 RnSFCPU Supported Functions

Operating procedure

Set the each item and click the [OK] button.

For the setting items when "Function Block" or "FB File" is selected as data type, refer to the following section.

☞ Page 214 Creating function blocks

For the setting items when "Function" is selected as data type, refer to the following section.

☞ Page 217 Creating functions

Precautions

- When function block and function data are added by the above method, the FB file and FUN file will be in the unconverted state.
- Once the FB file and FUN file is changed from the converted state to the unconverted state, all programs in which the function blocks and functions are used will be in the unconverted state.
- SFC data and program blocks written in programming languages except for SFC (Ladder, ST, FBD/LD) cannot exist in a same program file.

Editing projects

Changing a data name

Operating procedure

1. Select the data of which data name is to be changed on the Project view.
2. Select [Project] ⇨ [Data Operation] ⇨ [Rename Data]/right-click and select [Rename Data] from the shortcut menu.
3. Change the data name and press the Enter key.

Copying/pasting data

Utilize the data of project being edited or other projects.
The elements can be pasted only to the place where is able to create a same type of data.
If the module type is different between the copy source and the copy destination, the data can not be pasted.

Operating procedure

1. Select the data of the copy source on the Project view.
2. Select [Project] ⇨ [Data Operation] ⇨ [Copy Data] ( )/right-click and select [Copy Data] ( ) from the shortcut menu.
3. Select the target folder (one-level upper hierarchy of the copy source data) on the Project view or other project.
4. Select [Project] ⇨ [Data Operation] ⇨ [Paste Data] ( )/right-click and select [Paste Data] ( ) from the shortcut menu.

If the same data name exists in the folder where the data is to be pasted, the pasted data is named automatically.

Point

If the module type is different between the copy source and the copy destination, it can be utilized by opening each editor and copy the corresponding data.

Precautions

- The pasted program will be in the unconverted state.
- The structures/function blocks used in the global label cannot be copied.
  If the same structure/function block name does not exist in the folder where the data is to be pasted, the pasted data will be an undefined data type.
- Pasting of global label data is canceled when the maximum number of labels (20480) is exceeded. Adjust the number of global labels in the copy destination and the copy source, and retry pasting data.

■Program file

When copying a program file, the program blocks under the selected program file will be copied.
When pasting data to other project, the common device comments can also be copied by setting the following option.

- [Tool] ⇨ [Options] ⇨ "Edit" ⇨ "Copy" ⇨ "Operational Setting"

Deleting data

Delete the data of the project that is opened.

Operating procedure

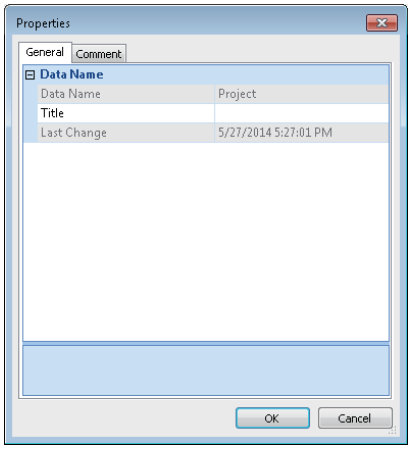
1. Select the data to be deleted on the Project view.
2. Select [Project] ⇨ [Data Operation] ⇨ [Delete Data]/right-click and select [Delete Data] from the shortcut menu.

Property

Display the properties of data such as folders, parameters, and programs.
Titles and comments can be added to each data.

Window

Select a data on the Project view, then [Project] ⇨ [Data Operation] ⇨ [Properties] ( )/right-click and select [Properties] ( ) from the shortcut menu.

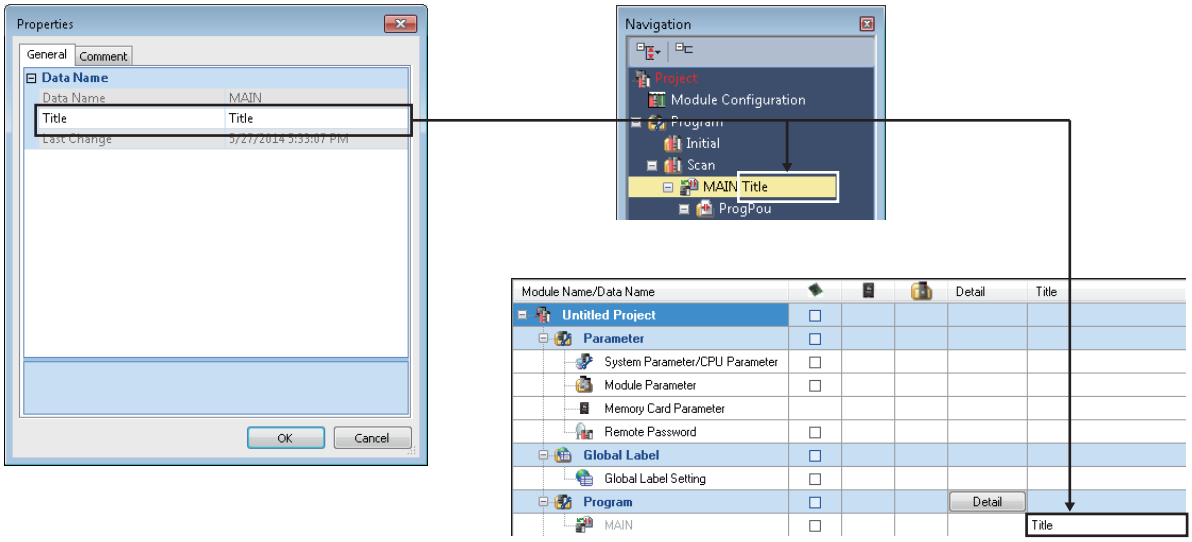


Operating procedure

Set the items and click the [OK] button.

Titles

The title set to a data is displayed with a data name on the Navigation window.
The title is also displayed on the "Online Data Operation" screen when writing/reading data to/from the programmable controller.

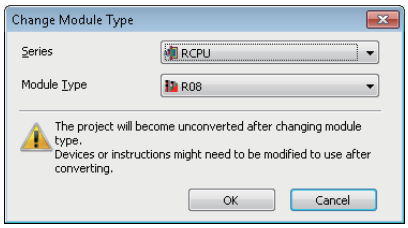


3.4 Changing Module Type of Project

Change the module type of the project being edited to other module type.
When a revision has been registered in the project revision, the revision will be succeeded after the change of module type.
FX5CPU does not support this function.

Window

[Project] ⇨ [Change Module Type]



Operating procedure

1. Select the module type to be changed, and click the [OK] button.
2. Compare the project after the change with the one before the change to confirm the changed contents with the Project Verify function.
3. Edit each data according to the module type after a change.

Precautions

- The changed module type cannot be restored. Save the project data in advance before changing the module type. Note that, the project after the change will be in unsaved state.
- When module labels of CPU module are used, the module labels before changing the module type are deleted, and the module labels after changing the module type are added. Therefore, the program may be required to be corrected after the module type is changed.

Series combinations of which module type is changeable

○: Changeable, △: Changeable (restricted), ×: Not changeable
For the restricted description, refer to the displayed message.

Series before change	Series after change			
	RCPU	RCPU(RnPCPU)	RCPU(RnSFCPU)	RCPU(RnENCPU)
RCPU	○	○	○	△
RCPU(RnPCPU)	○	○	○	△
RCPU(RnSFCPU)	×	×	○	×
RCPU(RnENCPU)	△	△	△	○

Point

To utilize the data to the different module type of CPU module, open each editor and copy the corresponding data.

Considerations for series combinations

■Changing a module type from RnENCPU to a CPU module that occupies a single slot

If the "I/O Assignment Setting" in the [I/O Assignment] tab of the system parameter has been configured, the network part, corresponding to the host CPU after changing the module type, will be the unplaced status. Therefore, the module type is changed to 'RJ71EN71' which is the same network type module.

■Changing a module type from a CPU module that occupies one slot to RnENCPU

If the "I/O Assignment Setting" in the [I/O Assignment] tab of the system parameter has been configured, a network module 'RJ71EN71' except for "E+E"/"Ethernet(Q series compatible)" needs to be placed on the right side of the CPU slot in the host CPU. After changing the module type, the 'RJ71EN71' will be in an unplaced status, and a CPU extension module '_RJ71EN71' which is the same network type will be placed.

3.5 Verifying Projects

Verify a open project with the data of other project.

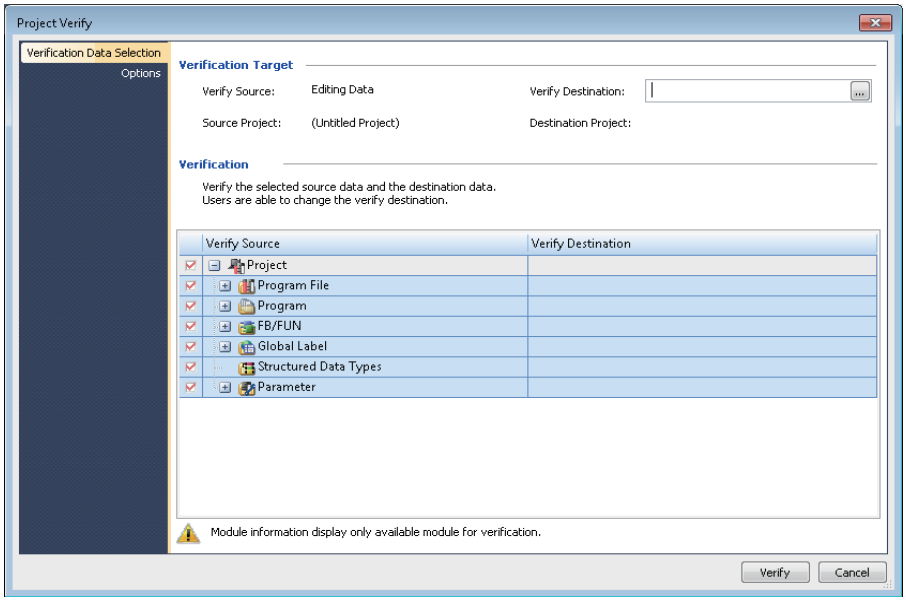
Use this function to check whether the program contents are consistent, or the modifications are reflected etc.

When a user information is registered to the project, the user authentication is required. (☞ Page 306 Logging on to CPU module)

Performing verification

Window

[Project] ⇨ [Project Verify]



Operating procedure

1. Specify the project to "Verify Destination" in [Verification Data Selection].
2. Select the data to be verified, and click the [Verify] button.

Point

The target project file can be specified by dragging and dropping it.

Precautions

- Verification with GX Works2/GX Developer project cannot be performed.
- Projects to which the security is set can be verified when both verify source data and verify destination data are not read-protected.

Parameters

■Verification levels

Select the parameter verification level from [Options] tab on the "Project Verify" screen.

Checking verification result

Check the details of mismatched data on the "Verify Result" screen.
Updated verify source data is displayed in the verification results.
Therefore, the verification result of updated verify source data can be checked without performing the Project Verify function again after mismatched data is modified.

Operating procedure

1. Select and double-click the row of data to be displayed in detail on the "Verify Result" screen.
2. Select [Find/Replace] ⇨ [Next Unmatched] ( )/[Previous Unmatched] ( ).

Point

Font color, background color, and fonts can be changed.
☞ Page 48 Checking and Changing Colors and Fonts

Precautions

For verification of program files, the verification results may mismatched because a difference occurred to the conversion results if the version of GX Works3, which were used for converting the program files of the verify destination, was different from the one of the verify source. In this case, convert all programs in both the verify source and the verify destination with the same version of GX Works3, then perform the verification again.

Ladder programs

The details of verification target ladder program is displayed by double-clicking the displayed program on the "Verify Result" screen.
The display format of editor/list can be selected from the pull-down list on the "Verify Result" screen.

■Detailed display (editor format)

The cursor is jumped to the corresponding instruction on the ladder editor by double-clicking a instruction in a ladder diagram.

Precautions

- When the instructions which cannot be displayed in the editor format (such as NOP) mismatched, it is displayed as it had matched. (Only "Mismatch" is displayed in the status bar). Check it by displaying in a list format.
- A background color is displayed in instruction units. However, it may be displayed to a cell that only a horizontal line is written on a ladder block. (Example: next cell of an input argument (BOOL value) for function block)

■Detailed display (list format)

- The cursor is jumped to a corresponding instruction on the ladder editor by double-clicking an instruction on the list.
- When the program that contains the inline structured text program is verified, "STB" is displayed on the row of the inline structured text.
  The cursor is jumped to the corresponding row of the inline structured text by double-clicking the "STB".
- When the program which includes functions/function blocks is verified, the result is displayed as follows:
  Function: A row of "*;FUN BLK START" to "*;FUN BLK END"
  Function block: A row of "*;FB BLK START" to "*;FB BLK END"

ST programs

■Detailed display

The cursor is jumped to a corresponding row on the ST editor by double-clicking a row on the verification result.

FBD/LD programs

The details of verification target FBD/LD program is displayed in a list format by double-clicking the displayed program on the "Verify Result" screen.

It is not available to display the details in a list format when verifying with a programmable controller or an FBD/LD program which has been read from a CPU module without converting all program.

Precautions

The verification results may mismatched because a difference occurred to the position information of elements if the version of GX Works3, which were used for converting the program files of the verify destination, was different from the one of the verify source. In this case, edit, convert, and save the programs in both the verify source and the verify destination with the same version of GX Works3, and perform the verification again[*1].

*1 When verifying with a programmable controller, perform the "Write to PLC" or "Online Program Change" in advance.

■Detailed display (list format)

- The cursor is jumped to a corresponding instruction on the FBD/LD editor by double-clicking an instruction on the list.
- When the program which includes functions/function blocks is verified, the result is displayed as follows:
  Function: A row of "*;FUN BLK START" to "*;FUN BLK END"
  Function block: A row of "*;FB BLK START" to "*;FB BLK END"

SFC programs

The details of the verification target SFC program is displayed in a list format by double-clicking the displayed program on the "Verify Result" screen.

It is not possible to verify in a Zoom and display the details in a list format when verifying with a programmable controller or the SFC program which has been read from a CPU module without converting all program.

Precautions

- Even if the SFC diagram displayed for the verification destination looked totally same as the one for the verification source, the verification results may mismatched because a difference will occur to the internal data depending on the creation procedure.
- Block information is excluded from the target of verification.

Parameters

The verification results may mismatched when the same profiles are not registered to the verify destination and the verify source

The details of verification target parameter are displayed in a table format by double-clicking the displayed parameter on the "Verify Result" screen.

■Detailed display (table format)

The cursor is jumped to a corresponding setting items on the parameter editor by double-clicking a parameter in a table.

Exporting data to file

Export the data displayed on the "Verify Result" screen to a file.

Operating procedure

Select [Edit] ⇨ [Export to File] ( ).

3.6 Project Revision

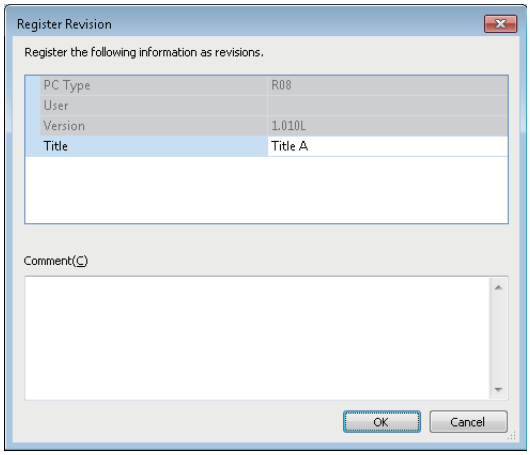
Register the project revision and manage the revision information.
The old project status can be restored using the registered revision information.

Registering revision information

Back up the project and register it with adding the revision information.

Window

[Project] ⇨ [Project Revision] ⇨ [Register Revision]



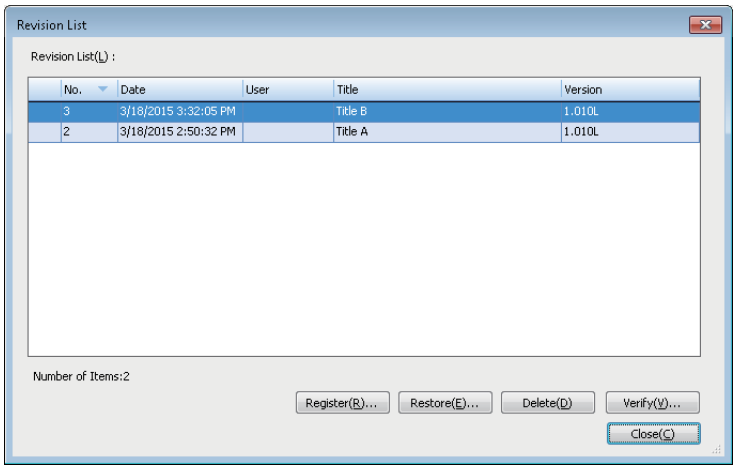
Operating procedure

Set the items and click the [OK] button.

Displaying revision list

Window

[Project] ⇨ [Project Revision] ⇨ [Revision List]



Select and right-click on the revision information, then select [Details] in the shortcut menu. The "Detailed Revision Information" screen is displayed, so check the information such as the comments entered at the time of registration.

Registering revision

Click the [Register] button to register the project revision.

Restoring revision information

Restore the registered revision information.

Operating procedure

1. Select the revision to be restored, and click the [Restore] button.
2. Click the [OK] button on the "Restore Revision" screen.

After restoring the project, ✅ is displayed at the head of restored revision information on the "Revision List" screen.

Precautions

- Before restoring a project, register the revision of the project being edited. If any of revision registered before is restored and the project is saved without registering the revision of the project being edited, the project before the restoration will be overwritten.
- The added/changed/deleted user information is not restored if the revision is restored.

Deleting revision information

Delete the registered revision information.

Operating procedure

Select the revision to be deleted, and click the [Delete] button.

Verifying revision

Check the revision information against revision information registered before or the project being edited.

Operating procedure

Select the revision to be verified, and click the [Verify] button.

By selecting only one revision, the selected revision information is checked against the project being edited. To verify between revisions registered before , select two revisions.

The operation method of the verification result is the same as the Project Verify function. For more details, refer to the following section.

☞ Page 78 Checking verification result

3.7 Managing Profile

Manage the registration status of profiles.
Profiles are the data in which the connected device information (such as model names) is stored.
The profiles are managed by each personal computer, and shared within GX Works3 and MELSOFT Navigator. Therefore, profiles registered in GX Works3 are applied to MELSOFT Navigator.
Before registering/deleting profiles, log on the personal computer as the user having the administrator authority, and close the project in advance.

Registration

Register profiles in GX Works3.

Operating procedure

1. Select [Tool] ⇨ [Profile Management] ⇨ [Register].
2. Select the file on the "Register Profile" screen, and click the [Register] button.

Precautions

Profiles are the compressed file (such as *.zip, *.ipar, and *.cspp).
Register profiles without decompressing files.

Deletion

Delete the registered profiles.

Operating procedure

1. Select [Tool] ⇨ [Profile Management] ⇨ [Delete].
2. Select the module name to be deleted on the "Profile Delete" screen, then click the [Delete] button.

4 CREATING MODULE CONFIGURATION DIAGRAM AND SETTING PARAMETERS

In GX Works3, parameters of a programmable controller can be set on the Module Configuration Diagram, as if to configure the actual system.
The parameters can be configured on the Project view in the same manner as GX Works2.

About Module Configuration Diagram

By using the Module Configuration Diagram, the following operations can easily be performed.

Item	Reference
Display an actual programmable controller system configuration visually	Page 85 Creating Module Configuration Diagram
Set parameters for modules	Page 89 Setting parameters on the Module Configuration Diagram
Input start XY in batch*1	Page 92 Inputting start XY in batch
Input default points in batch*1	Page 92 Inputting default points in batch
Check a power supply capacity and I/O points*1	Page 92 Checking power supply capacity and I/O points
Check a system configuration	Page 92 Checking system configurations

*1 Not supported by FX5CPU.

About parameter settings

There are two methods for setting parameters; use the Module Configuration Diagram or use the Project view.
For the operation methods, refer to the following sections.
☞ Page 89 Setting parameters on the Module Configuration Diagram, Page 93 Setting Parameters

4.1 Creating Module Configuration Diagram

Place the module parts (object) with the same configuration as an actual system configuration on the Module Configuration Diagram.

The Module Configuration Diagram of GX Works3 can be configured within the range that can be managed by the CPU module of a project.

Editor configuration of Module Configuration Diagram

Point

Display settings can be changed by setting the following option.

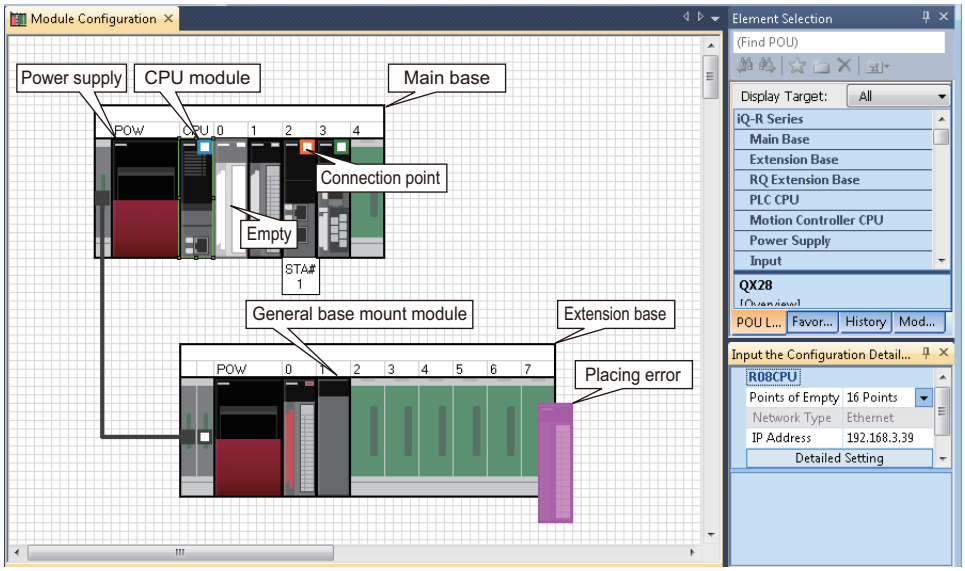
[Tool] ⇨ [Options] ⇨ "Other Editor" ⇨ "Module Configuration Diagram"

Window

Double-click "Module Configuration" on the Project view.

●Toolbar

●Module Configuration Diagram



Displayed items

Screen name	Item	Description
Module Configuration Diagram	iQ-R series	An object of MELSEC iQ-R series module supported by GX Works3.
	FX5 series	An object of MELSEC iQ-F FX5 series module supported by GX Works3.
	Q series	An object of Q series module supported by GX Works3.
	General module	An object of module for which the start XY and point can be set arbitrarily by each module type. The same points as empty slot points can be assigned as an initial value. Select this when using a module which is not displayed in the Element Selection window.
	Figure (Straight Line, Rectangle, Ellipse, Text Box)	An object that is used when adding information such as description on the Module Configuration Diagram.
	Connection line	An object of a bus cable to be connected to a base unit.
	Connection point	A point to be connected with a connection line. The connection point located at the upper right on the module is used when importing a project using MELSOFT Navigator. It is not used in GX Works3.
	Module status setting (empty)	A module to be used when operating system by placing modules on the Module Configuration Diagram without mounting actual modules. The module placed on the Module Configuration Diagram and its related parameters will also be set as "(Empty)". (Reserved status)
	Object placement error	The module object which is not placed properly is highlighted.
Element Selection window		A list of objects which can be used on the Module Configuration Diagram.
Input the Configuration Detailed Information window		A window to enter the information such as start XY or station number of the module placed on the Module Configuration Diagram.

Placing objects

Placing module objects

Operating procedure

1. Select a main base from the Element Selection window, and drag and drop it onto the Module Configuration Diagram.
2. Select a module from the Element Selection window, and drag and drop it on the base unit placed on the Module Configuration Diagram in the Step 1.

The available locations are highlighted while dragging the module.

Point

Object names of module objects can be changed on the "Properties" screen. It will be easy to distinguish a module from one having a same module name by changing an object name.

Precautions

- GOT2000/GOT1000 series, general modules, image diagrams, and link files supported by MELSOFT Navigator are not supported.
- Only bus cables are available. Network connection and serial connection are not available.
- For FX5CPU, bus cables are not supported. In addition, a base unit (slot) is not required. Select the module, and drag and drop it on the Module Configuration Diagram directly.

■Moving placed modules

If a module is removed from the main base or extension base, the object information such as start I/O and parameter information is retained. Therefore, when the removed module is placed on the base unit again, the retained parameter information will be set automatically.

The module removed from the Module Configuration Diagram is deleted from the I/O Assignment Setting screen.

For FX5CPU, the setting cannot be fixed in the state where a module has been removed from the CPU module.

■Deleting placed modules

When a module is deleted, the module information displayed on the Project view after fixing parameters is deleted as well.

For a single CPU system, a CPU module object cannot be deleted.

As for a multiple CPU configuration, the host CPU cannot be deleted.

■Copying and pasting modules

The parameter information of the copy source is succeeded to the pasted module.
If the CPU module placed on the base unit in a multiple CPU configuration is copied and pasted, the CPU number will be unspecified status.
The copied objects can be pasted to the Module Configuration Diagram in other project.

Placing symbol objects

The symbols and text boxes can be placed on the Module Configuration Diagram.
The symbols placed on the Module Configuration Diagram are not written to a CPU module. They are saved only in the project.

Operating procedure

Select a figure in "Figure" in the Element Selection window, and drag and drop it onto the Module Configuration Diagram.
Change the character and color of a symbol object on the "Properties" screen.

Changing model name of CPU module

The model name of the CPU module placed on the Module Configuration Diagram can be changed.
RCPU does not support it.

Operating procedure

1. Select the placed CPU module. Then, right-click and select [Change CPU Model Name] from the shortcut menu.
2. Select the model name after the change on the "Change CPU" screen.

Module status setting (empty)

Set this when operating a system without mounting actual modules.
The module is displayed in a pale color.
FX5CPU does not support this setting.

Operating procedure

Select the module to be set to empty, and [Edit] ⇨ [Module Status Setting (Empty)].

Reading module configuration from actual system

The Module Configuration Diagram can be configured by the information read from CPU module.
FX5CPU does not support it.

Operating procedure

Select [Online] ⇨ [Read Module Configuration from PLC].

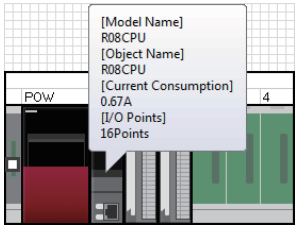
Precautions

Module configuration can not be read to the extension base units where its number is incorrectly set.

Checking module information

Check on a balloon help

Place the cursor on the object (such as module and symbol) to display a balloon help on the Module Configuration Diagram.

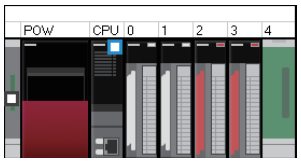


Check model names on module objects

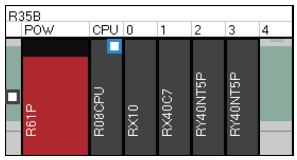
Operating procedure

Select [Edit] ⇨ [Display Module Information] ( ).

Normal display

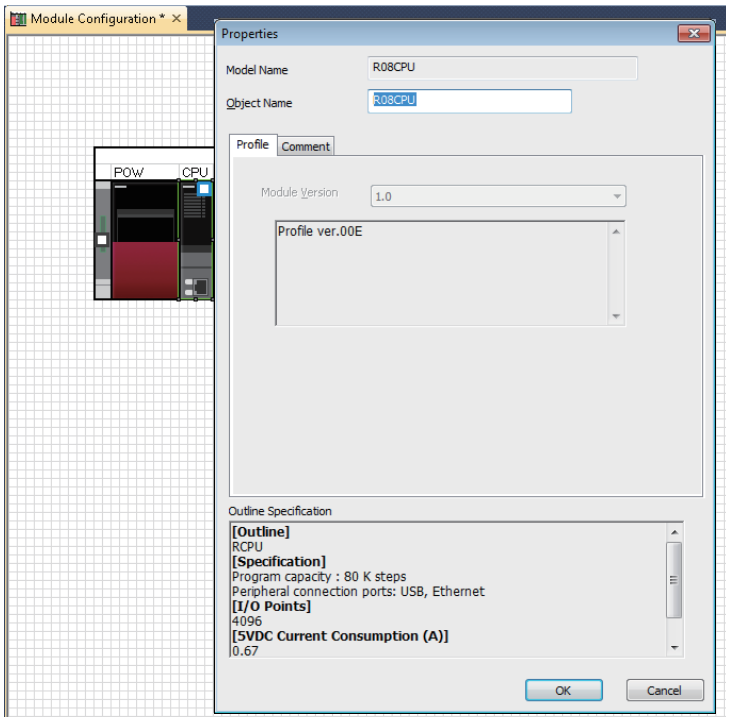


Module information display



Check on the "Properties" screen

A model name, object name, and outline of a specification can be checked on the "Properties" screen.
Additionally, a comment can be entered.



Operating procedure

Select a module on the Module Configuration Diagram, right-click and select [Property] from the shortcut menu.

Setting parameters on the Module Configuration Diagram

Set the parameter of the control CPU and module on the Module Configuration Diagram.

Operating procedure

1. Select the module to edit the parameters.
2. Set the items on the Input the Configuration Detailed Information window.
3. Select [Edit] ⇨ [Parameter] ⇨ [Fix] ( ).

When setting details of the module, click the [Detailed Setting] button to display the parameter editor.

■Fixing parameters

After editing an object on the Module Configuration Diagram, the parameters of the edited object will be undefined status. The operations that the parameters become undefined are as follows:

- Adding or deleting a module
- Removing a module from the slot, or placing the removed module on the slot
  For FX5CPU, removing/connecting a module from/to the CPU module
- Cutting and pasting modules
- Connecting modules
- Performing undo/redo
- Changing property of a module
- Performing the Start XY Batch Input function
- Performing the Default Points Batch Input function
- Setting the Module status setting (empty)
- Operations on the Input the Configuration Detailed Information window

The fixed parameters are reflected to the following screens.

- Project view (For a new module information, the module information is added to the Project view.)
- Parameter editor
- I/O Assignment Setting screen

■List of changed location of start XY number

When the start XY number of a module is changed on the Module Configuration Diagram, the list of data which will be affected by the change are displayed on the "Module Start I/O No. Related Area" window.

FX5CPU does not support it.

Check the display content, and correct the corresponding data.

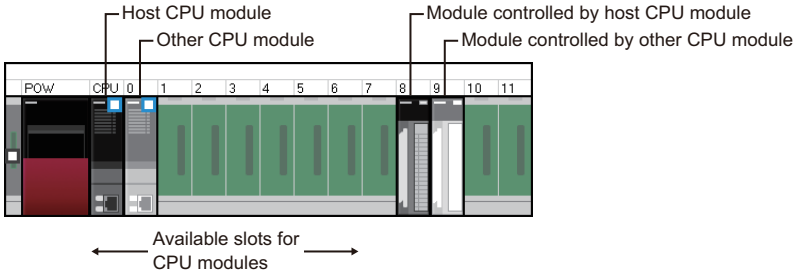
- Target data: Program, FB program, system parameter, CPU parameter, module parameter, global label, structure, module label
- Target device: The start XY devices (X, Y, DX, DY, U, G) specified in the arguments of FROM (P), TO (P), DFRO (P), or DTO (P) instructions

Multiple CPU configuration

FX5CPU does not support it.

Display of Multiple CPU configuration

The multiple CPU configuration can be set on the Module Configuration Diagram.



Item	Description
Host CPU module	For a multiple CPU configuration, only one project which is set as a host CPU can be set.
Other CPU module	Set the CPU module which is not set as a host CPU to other station. The module is displayed in a pale color, same as when the module state setting (empty) is set.
Module controlled by the host CPU module	A module that is controlled by the CPU module set as a host CPU.
Module controlled by the other station CPU module	A module that is controlled by the CPU module set as other station. The module is displayed in a pale color, same as when the module state setting (empty) is set.
Available slots for CPU modules	Up to four CPU modules can be placed for CPU slot and slot 0 to 6. The placed CPU modules will be in order from CPU No.1 to 4, from left to right.

Setting multiple CPU configuration

In the multiple CPU configuration, place more than two CPU modules on the base unit.

When changing the control CPU of the module, use the Input the Configuration Detailed Information window.

The following explains how to change a single CPU configuration to a multiple CPU configuration, and vice versa.

■When changing a single CPU configuration to a multiple CPU configuration

Drag and drop the CPU module from the Element Selection window with a single CPU configuration (one CPU is placed on the base unit), and place the second CPU module on the base unit.

■When changing a multiple CPU configuration to a single CPU configuration

In a multiple CPU configuration (two or more CPU modules are placed on a base unit), delete (or remove) the CPU modules from the slot and remain only one module.

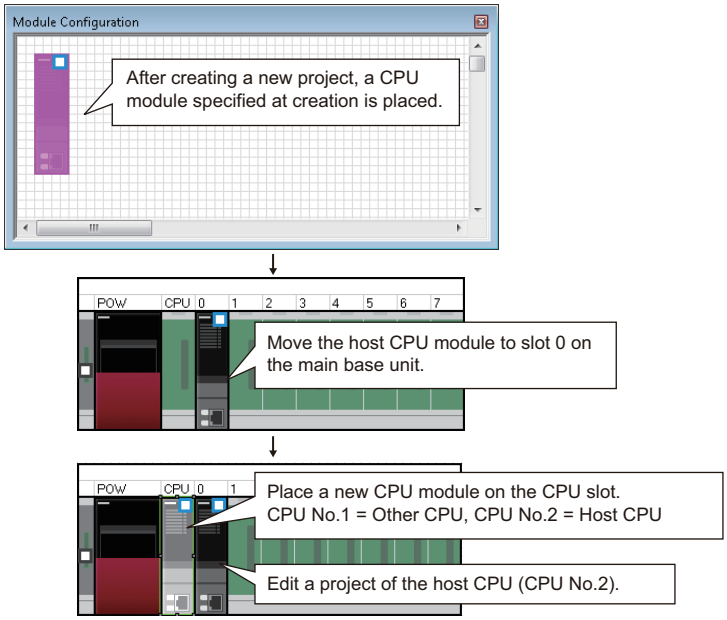
Changing CPU number of the CPU module set as a host CPU

For a multiple CPU configuration, the CPU number of a CPU module is set in order from left on the base unit.

When changing the CPU number, change the position of the CPU module.

■Operation for single CPU system

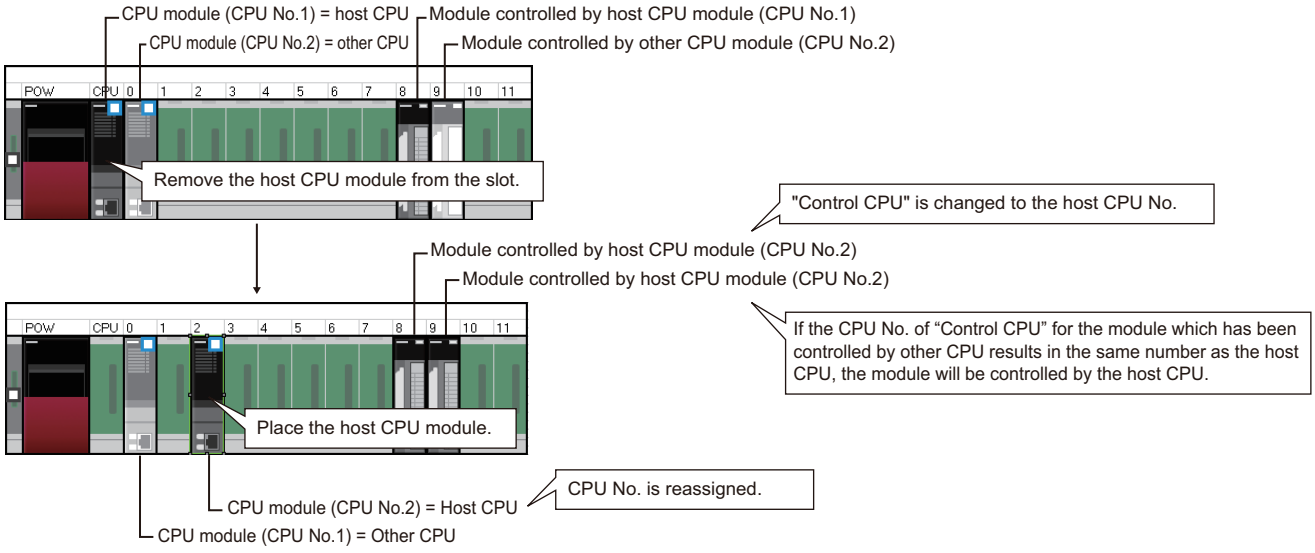
Move the CPU module placed on the CPU slot to an empty slot, and place a new CPU module on the empty slot.


Module Configuration
After creating a new project, a CPU module specified at creation is placed.
POW
CPU
0
1
2
3
4
5
6
7
Move the host CPU module to slot 0 on the main base unit.
Place a new CPU module on the CPU slot.
CPU No.1 = Other CPU, CPU No.2 = Host CPU
Edit a project of the host CPU (CPU No.2).

■Operation for multiple CPU configuration

Exchange the CPU module on the CPU slot and the CPU module on other slot.

Since the parameter information of the modules removed from the base unit is retained, the retained information is succeeded when the removed modules are placed again.


CPU module (CPU No.1) = host CPU
CPU module (CPU No.2) = other CPU
Module controlled by host CPU module (CPU No.1)
Module controlled by other CPU module (CPU No.2)
POW
CPU
0
1
2
3
4
5
6
7
8
9
10
11
Remove the host CPU module from the slot.
"Control CPU" is changed to the host CPU No.
Module controlled by host CPU module (CPU No.2)
Module controlled by host CPU module (CPU No.2)
If the CPU No. of “Control CPU” for the module which has been controlled by other CPU results in the same number as the host CPU, the module will be controlled by the host CPU.
Place the host CPU module.
CPU No. is reassigned.
CPU module (CPU No.2) = Host CPU
CPU module (CPU No.1) = Other CPU

Inputting start XY in batch

Set the start XY of each module in order of the slot number in a batch.
FX5CPU does not support it.

Operating procedure

Select [Edit] ⇨ [Start XY Batch Input] ( ).

Inputting default points in batch

Change the points, points occupied by empty slot, and empty slot side points (for a module occupies two or more slots) in all the modules and empty slots on the Module Configuration Diagram to default in a batch.
FX5CPU does not support it.

Operating procedure

Select [Edit] ⇨ [Default Points Batch Input].

Checking power supply capacity and I/O points

Check if the power capacity and I/O points in the configuration on the Module Configuration Diagram exceed the upper limit.
FX5CPU does not support it.

Operating procedure

1. Select [Edit] ⇨ [Check] ⇨ [Power Supply Capacity and I/O Points] ( ).
2. Check the result displayed on the Result of Power Supply Capacity and I/O Points Check window.

Checking system configurations

Check if the layout of each module on the Module Configuration Diagram is proper. The result is displayed on the Output window.

Operating procedure

1. Select [Edit] ⇨ [Check] ⇨ [System Configuration] ( ).
2. Check the result displayed on the Output window.

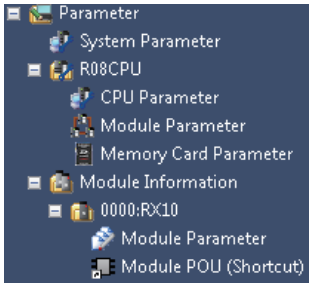
Operation for Write to PLC/Read from PLC

Only parameters of the modules in the information of Module Configuration Diagram can be read/written.
After reading data from a programmable controller, the coordinate position of the modules on the Module Configuration Diagram are displayed by default status.
The information of symbols is deleted.

4.2 Setting Parameters

To operate a programmable controller, setting parameters for the modules is required.

The following explains the settings of parameters on the Project view.

Project view	Item	Description	Reference
	System Parameter	Set the items which are required to configure the system such as a module configuration. The 'system' here indicates the system configured with a main base unit, extension base unit, and RQ extension base unit connected with the extension cable for RCPU, and the system configured with a module and adapter connected to the module for FX5CPU.	Page 95 Setting system parameters
	(Parameter of control CPU)	Set the operation of the function of a CPU module itself. This includes the file setting, memory/device setting, and program setting.	Page 96 Setting parameters of control CPU
	Module Information	Parameters that is set to an I/O module and Intelligent function module. This includes initial values and refresh settings of each module. There are two kinds of parameters; 'Module Parameter' and 'Module Extended Parameter'. • Module parameter: A parameter that is set to an I/O module and Intelligent function module. The initial setting values and refresh settings are included. • Module extended parameter: A parameter that is set to a specific Intelligent function module. This parameter is read/written separately from module parameters.	Page 97 Setting parameters of I/O module and Intelligent function module

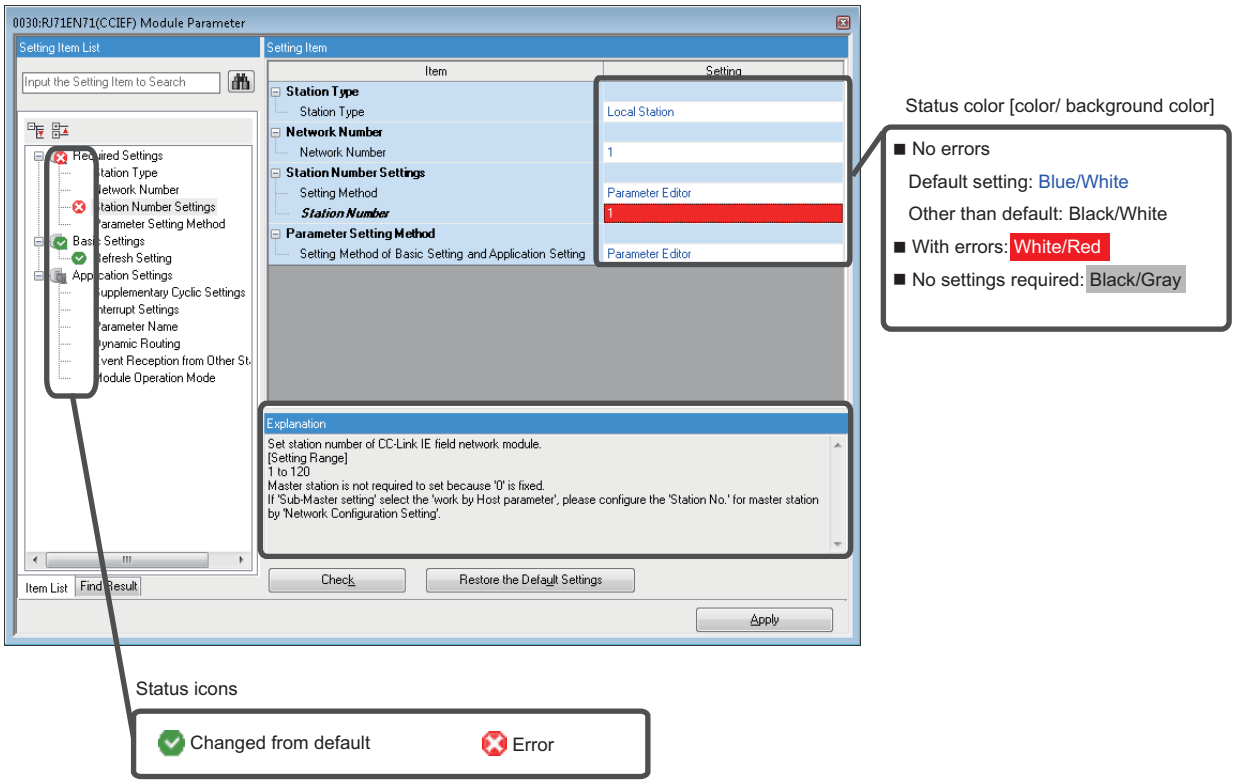
Common operations of parameter settings

Operation for parameter editor

The parameter items of the system parameter, control CPU parameter, and module information are displayed corresponding to their purpose.

Set the items refer to the information displayed in "Explanation".

Enter the keyword to search from the setting items and "Explanation".



Checking parameters

There are two methods for checking parameters:

- Check if there is an input error on the parameter editor.
  Click the [Check] button on the parameter editor.
- Check if there is an error in the parameter set in the project.
  Select [Tool] ⇨ [Check Parameter].

Setting system parameters

Set the parameters such as I/O Assignment Setting, Multiple CPU Setting, and Synchronization Setting within the Modules that affect whole system.

FX5CPU does not support the Multiple CPU Setting and Synchronization Setting within the Modules.

For details on the setting items of the parameters, refer to the following manuals.

- I/O Assignment Setting: MELSEC iQ-R Module Configuration Manual, MELSEC iQ-F FX5 User's Manual (Application)
- Multiple CPU Setting: MELSEC iQ-R CPU Module User's Manual (Application)
- Synchronization Setting within the Modules: MELSEC iQ-R Inter-Module Synchronization Function Reference Manual

Point

System parameters of other project can be utilized.

Click the [System Parameter Diversion] button, and select the project file to be used.

Precautions

When configuring a multiple CPU configurations using RnENCPU, the configuration cannot be changed to a single CPU configuration on the I/O Assignment Setting screen. To change it, replace the module on the Module Configuration Diagram.

Setting parameters

Operating procedure

1. Double-click "Parameter" ⇨ "System Parameter" on the Project view.
2. Select the items to be set in the [I/O Assignment] tab, [Multiple CPU Setting] tab, or [Synchronization Setting within the Modules] tab.
3. Set the items and click the [OK] button.

■When module is deleted in I/O Assignment Setting

The parameter of the deleted module is displayed as "Unset" on the Project view. To restore the state of the parameter (to display in I/O Assignment Setting again), select the mount position on the "Properties" screen.

■When I/O Assignment Setting is read from the read mounting status

Mounting status (Number of CPU modules, the I/O Assignment Setting (module name/point), Base/Power/Extension Cable Setting) cannot be read properly to the extension base units where its number is incorrectly set.

Point

Mounting status can be read to the I/O Assignment Setting by clicking the [Read Mounting Status] button.

Setting parameters of control CPU

Set the parameters related to CPU module (the host CPU for a multiple CPU configuration).

For details on the setting items of the parameters, refer to the following manuals.

- CPU parameters, memory card parameters: MELSEC iQ-R CPU Module User's Manual (Application), MELSEC iQ-F FX5 User's Manual (Application)
- Module parameters: MELSEC iQ-R Ethernet User's Manual (Application), MELSEC iQ-F FX5 User's Manual (Application)

Setting parameters

Operating procedure

1. Double-click "Parameter" ⇨ "(PLC type of the project)" ⇨ "CPU Parameter"/"Module Parameter"/"Memory Card Parameter" on the Project view.
2. Set the items on the parameter editor.
3. Click the [Apply] button or [OK] button.

Point

The setting screen can also be opened by double-clicking the object of a module on the Module Configuration Diagram.

Precautions

For the setting values of the parameters, use the characters in the Unicode Basic Multilingual Plane.

If the characters outside the Unicode Basic Multilingual Plane are specified, they may not be edited in the program.

- File Register Setting in File Setting: File Name
- Initial Value Setting in File Setting: Global Device Initial Value File Name
- Program Setting in Program Setting: Program Name
- FB/FUN File Setting in Program Setting: FB/FUN File Name

Setting parameters of I/O module and Intelligent function module

Set the parameters of MELSEC iQ-R series/Q series I/O module and Intelligent function module supported by GX Works3.
Set the switch settings and refresh settings of MELSEC-Q series module on the parameter editor.
For details of the items for parameters, refer to the user's manual of each module.

Setting parameters

Operating procedure

1. Select "Parameter" ⇨ "Module Information" on the Project view, and select [Project] ⇨ [Data Operation] ⇨ [Add New Module]/right-click and select [Add New Module] from the shortcut menu.
2. Select the items on the "Add New Module" screen.
3. Double-click the created parameter.

Setting network configuration and target devices

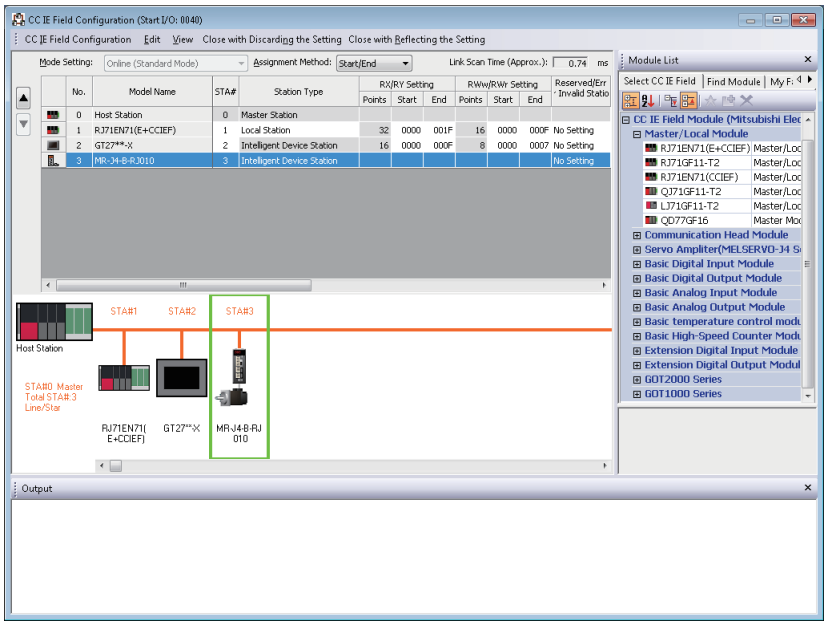
Set the network configuration and the target device of Ethernet, CC-Link IE Field, and CC-Link on the configuration setting screen.
For details on the settings, refer to the user's manual of each network.

Window

The configuration setting screen is displayed from one of the following module parameter items.

- Ethernet: External Device Configuration of Basic Settings
- CC-Link IE Field (master station)*1: Network Configuration Settings of Basic Settings
- CC-Link (master station)*1: Network Configuration Settings of Basic Settings

*1 Not supported by FX5CPU.



Point

For CC-Link IE Field Network, connected devices are detected automatically and also the information can be reflected to the configuration settings using the [Detect Now] button on the CC IE Field Configuration window.
For details, refer to the following manual.
📖 iQ Sensor Solution Reference Manual

■Property

Depending on the selected module, the image diagram can be changed on the "Properties" screen displayed by right-clicking and selecting [Property] from the shortcut menu.

Configuration applications, setting files, and manuals can be linked to the modules. Double-click the added module to open the linked application or file.

Checking refresh devices assigned to modules

Display the refresh devices assigned to each CC-Link module in a list.

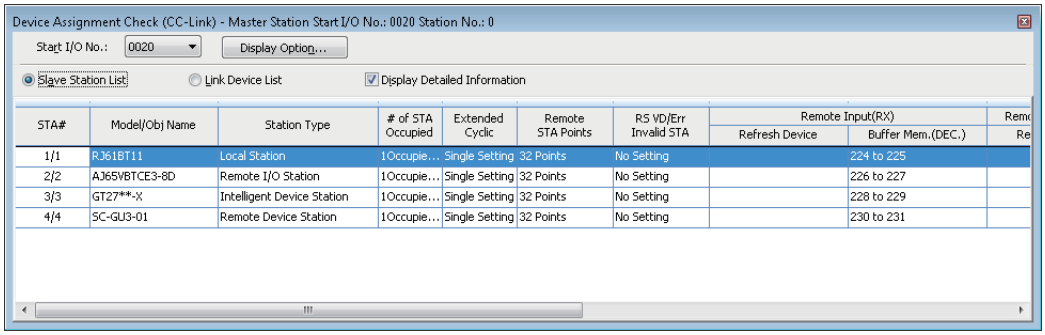
Set the CC-Link configuration from the parameter of CC-Link module in advance.

FX5CPU does not support it.

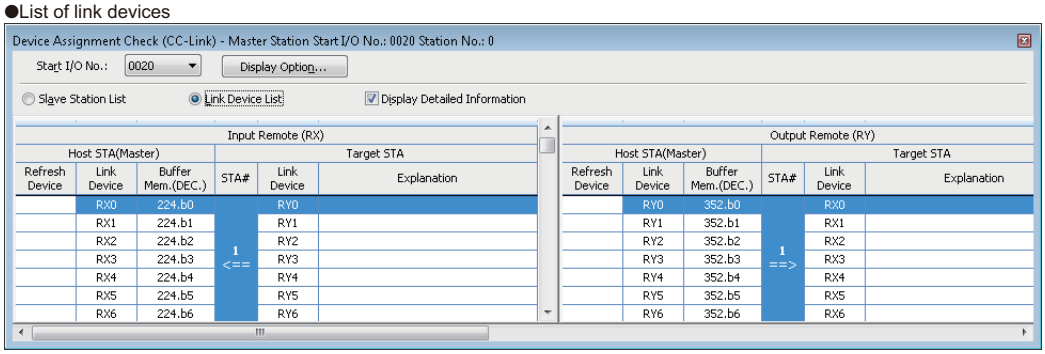
Window

[View] ⇨ [Docking Window] ⇨ [Device Reference] (DEV)

●List of slave stations


Device Assignment Check (CC-Link) - Master Station Start I/O No.: 0020 Station No.: 0
Start I/O No.: 0020
Display Option...
Slave Station List
Link Device List
Display Detailed Information
STA# | Model/Obj Name | Station Type | # of STA Occupied | Extended Cyclic | Remote STA Points | RS VD/Err Invalid STA | Remote Input(RX) Refresh Device | Buffer Mem.(DEC.)
1/1 | RJ61BT11 | Local Station | 1Occupie... | Single Setting | 32 Points | No Setting | | 224 to 225
2/2 | AJ65VBTCE3-8D | Remote I/O Station | 1Occupie... | Single Setting | 32 Points | No Setting | | 226 to 227
3/3 | GT27**-X | Intelligent Device Station | 1Occupie... | Single Setting | 32 Points | No Setting | | 228 to 229
4/4 | SC-GU3-01 | Remote Device Station | 1Occupie... | Single Setting | 32 Points | No Setting | | 230 to 231

●List of link devices


Device Assignment Check (CC-Link) - Master Station Start I/O No.: 0020 Station No.: 0
Start I/O No.: 0020
Display Option...
Slave Station List
Link Device List
Display Detailed Information
Input Remote (RX)
Host STA(Master): Refresh Device | Link Device | Buffer Mem.(DEC.)
Target STA: STA# | Link Device | Explanation
RX0 224.b0 1 <== RY0
RX1 224.b1 RY1
RX2 224.b2 RY2
RX3 224.b3 RY3
RX4 224.b4 RY4
RX5 224.b5 RY5
RX6 224.b6 RY6
Output Remote (RY)
Host STA(Master): Refresh Device | Link Device | Buffer Mem.(DEC.)
Target STA: STA# | Link Device | Explanation
RY0 352.b0 1 ==> RX0
RY1 352.b1 RX1
RY2 352.b2 RX2
RY3 352.b3 RX3
RY4 352.b4 RX4
RY5 352.b5 RX5
RY6 352.b6 RX6

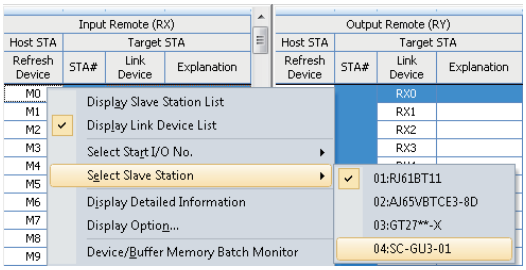
"Displayed Content for Model Name" and "Buffer Memory Display Format" can be selected by clicking the [Display Option] button.

Displayed items

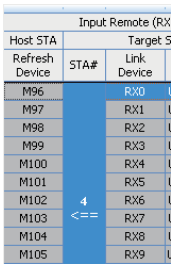
Item	Description
Start I/O No.	Select the start I/O number of master station or local station to be displayed on the list.
Slave Station List	Displays the information of the slave stations and the range of the refresh devices assigned to the own station. Multiple devices are displayed in accordance with the link refresh settings of the Module Parameter. When refresh devices are not assigned, the cells of the refresh device range will be blank.
Link Device List	Displays the assignment status of refresh devices on the own station and link devices on the target station. When link devices are not assigned, the cells of the link device range will be blank. The cells of "Explanation" will be blank when profile is not registered.

Point

- The arbitrary slave station can be displayed on top of the cell by right-clicking on the list of link devices and selecting [Select Slave Station] from the shortcut menu.



Select the station number 4 of the slave station.



The station number 4 is displayed at the top.

- The "Device/Buffer Memory Batch Monitor" screen is displayed by right-clicking a refresh device or buffer memory from the list of link devices and selecting [Device/Buffer Memory Batch Monitor] from the shortcut menu.

Checking/changing the number of Intelligent function module parameters

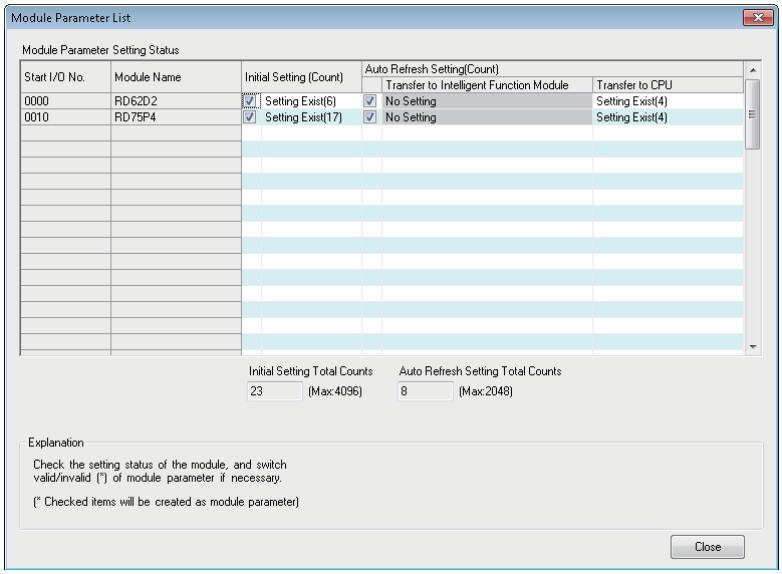
Display the setting information of the start XY addresses, initial settings, and auto refresh settings of Intelligent function modules in a list.

Intelligent function module parameters perform the following operations by writing parameters in a CPU module.

- Initial setting: The parameters of the individual Intelligent function module data are set as an initial setting.
  The data is registered to the CPU module parameters and written to the Intelligent function module automatically when the CPU module turns to the RUN state.
- Auto refresh: For the buffer memory of an Intelligent function module set to auto refresh, the data is written to/read from the specified devices automatically when the END instruction is executed on the CPU module.

Window

- [Project] ⇨ [Intelligent Function Module] ⇨ [Module Parameter List]
- Select "Parameter" ⇨ "Module Information" on the Project view, and right-click and select [Module Parameter List] from the shortcut menu.



■Enabling/disabling parameters

Set whether to enable/disable the initial setting and auto refresh as an intelligent function module parameter.

Since a CPU module limits the number of parameters that can be set for the mounted Intelligent function modules, check the setting information using this function, and enable/disable the parameters so that the number of set parameters is within the allowable range.

For details on the number of the parameter settings, refer to the manuals of respective Intelligent function modules.

Operating procedure

Set the items and click the [Close] button.

Item		Description
Initial Setting (Count)		Unselect the item if it is not set as an intelligent function module parameter. For a module with no initial settings, "-" is displayed.
Auto Refresh Setting (Count)	Transfer to Intelligent Function Module	Unselect the item if it is not set as an intelligent function module/CPU module parameter. For a module for which auto refresh is not set, "No Setting" is displayed.
	Transfer to CPU	
Initial Setting Total Counts		Displays the number of initial settings set as intelligent function module parameters.
Auto Refresh Setting Total Counts		Displays the number of auto refreshes set as intelligent function module parameters.

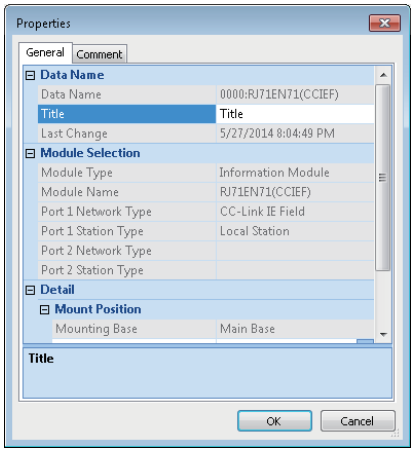
Checking property information

Check the setting information of Intelligent function module.

By this operation, the mounting slot number, start I/O number, and title (up to 32 characters) for RCPU, and the mounting position number and title (up to 32 characters) for FX5CPU can be changed.

Window

Select "Parameter" ⇨ "Module Information" ⇨ "(module name)" on the Project view, then select [Project] ⇨ [Data Operation] ⇨ [Properties] ( )/right-click and select [Properties] ( ) from the shortcut menu.



Operating procedure

Set the items and click the [OK] button.

Module dedicated menus for the parameter editor

Depending on the setting items of each module, the input format and device assignment method can be selected from one of the following menus.

- [Edit] ⇨ [IP Address Input Format] ⇨ [Decimal]/[Hexadecimal]
- [Edit] ⇨ [Device Assignment Method] ⇨ [Start/End]/[Points/Start]
- [Edit] ⇨ [Word Device Setting Value Input Format] ⇨ [Decimal]/[Hexadecimal]

Parameter interaction with MELSOFT Navigator

By using the parameter interaction function of MELSOFT Navigator, the parameter consistency can be ensured between MELSOFT Navigator and GX Works3.

The operation is performed with MELSOFT Navigator. For details, refer to the HELP of MELSOFT Navigator.

4.3 Other settings of Intelligent function module

The settings except for the parameter setting of the Intelligent function module can be configured with a module tool or a dedicated tool.

Setting by module tool

List of setting function

■MELSEC iQ-R series

Supported module	Function name	Reference
Analog input module	Offset/gain setting	📖 MELSEC iQ-R Analog-Digital Converter Module User's Manual (Startup)
Analog output module	Offset/gain setting	📖 MELSEC iQ-R Digital-Analog Converter Module User's Manual (Startup)
	Create wave output data	
Temperature input module	Offset/gain setting	📖 MELSEC iQ-R Channel Isolated Thermocouple Input Module/Channel Isolated RTD Input Module User's Manual (Startup)
Temperature control module	Temperature trace	📖 MELSEC iQ-R Temperature Control Module User's Manual (Startup)
Pulse I/O, positioning module	Preset	📖 MELSEC iQ-R High-Speed Counter Module User's Manual (Application)
	Positioning monitor	📖 MELSEC iQ-R Positioning Module User's Manual (Application)
	Positioning test	

■MELSEC iQ-F series

Supported module	Function name	Reference
Analog adapter module	Analog input - offset/gain setting	📖 MELSEC iQ-F FX5 User's Manual (Analog Control)
	Analog output - offset/gain setting	

■Q series

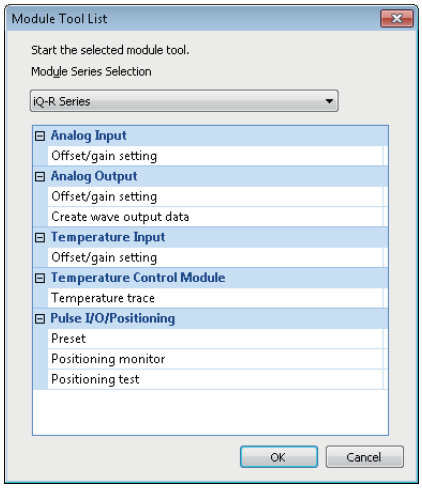
Supported module	Function name	Reference
Analog module	Offset/gain setting	📖 User's manual of the module used
	Q61LD two-point calibration setting	
	Q61LD default setting	
	Create wave output data	
Temperature input module	Offset/gain setting	
Temperature control module	Auto-tuning	
	Sensor correction function	
Counter module	Preset	

Displaying module tool list

The configuration functions can be performed on the "Module Tool List" screen.

Window

[Tool] ⇨ [Module Tool List]



Simple motion module setting tool

Set parameters and positioning data of simple motion module using the simple motion module setting tool.
For details on the operation methods and setting items, refer to the help of Simple Motion Module Setting Tool.

Operating procedure

1. Select "Parameter" ⇨ "Module Information" on the Project view, and select [Project] ⇨ [Data Operation] ⇨ [Add New Module].
2. Select the items on the "Add New Module" screen.
3. Double-click the added simple motion module setting.

Precautions

The simple motion module settings are saved in the project of GX Works3 when saving the GX Works3 project.

4.4 Predefined Protocol Support Function

Start the Predefined Protocol Support Function from GX Works3, and set the protocol and read/write data from/to a module.
For details, refer to the following manuals.

📖 MELSEC iQ-R Serial Communication Module User's Manual (Application)
📖 MELSEC iQ-R Ethernet User's Manual (Application)
📖 MELSEC iQ-F FX5 User's Manual (Serial Communication)
📖 MELSEC iQ-F FX5 User's Manual (Ethernet Communication)

Point

The files saved in GX Works2 can be read using the Predefined Protocol Support Function of GX Works3.

Starting and exiting predefined protocol support function

Start and exit the predefined protocol support function.

Starting Predefined Protocol Support Function

Operating procedure

1. Select GX Works3 menu [Tool] ⇨ [Predefined Protocol Support Function].
2. Set the items on the "Predefined Protocol Support Function" screen, and click the [OK] button.

Exiting Predefined Protocol Support Function

Operating procedure

Select [File] ⇨ [Exit] on the protocol setting screen.

4.5 Circuit Trace Function

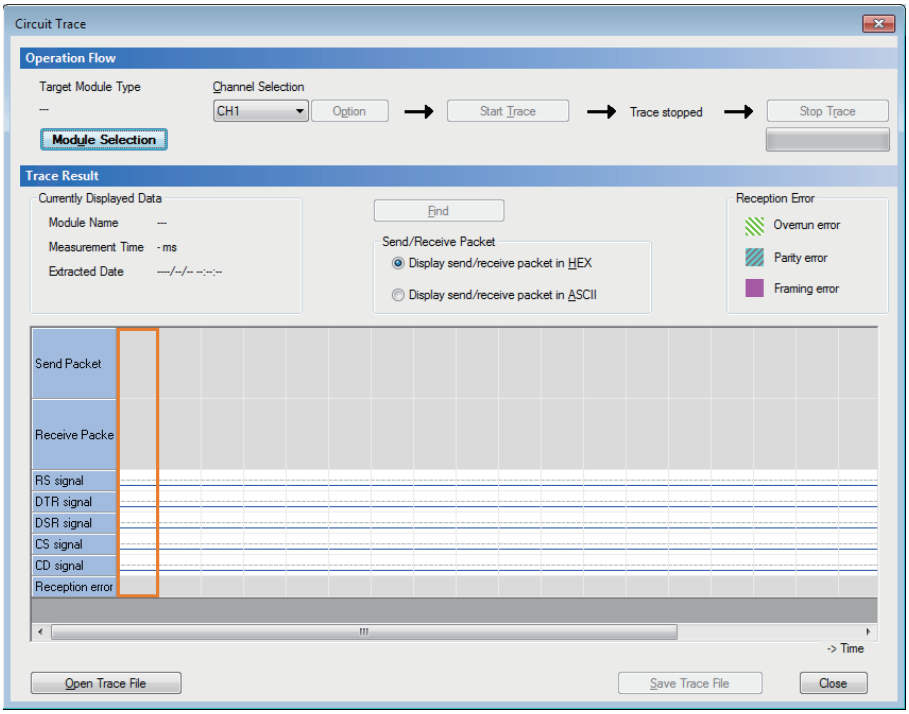
Trace the send/receive data and communication control signal between C24 and a target device.

For details, refer to the following manual.

MELSEC iQ-R Serial Communication Module User's Manual (Application)

Window

[Tool] ⇨ [Circuit Trace]



5 REGISTERING LABELS

This chapter explains the overviews and registration methods of labels.

5.1 About Labels

There are four types of label; labels registered on the label editor (global label and local label), a module label which is prepared as the dedicated label for each module (global label), and a system label which is interacted with MELSOFT Navigator.

Global labels can be registered as system label.

Type	Description	Creation method	Reference
Global label	Labels that can be used in all programs in a project. There are two types of global label; a global label that can be used either in standard programs or in safety programs, and a standard/safety shared global label that can be used in both programs.	Create them on the global label editor.	Page 107 Registering Labels
Local label	Labels that can be used in each program.	Create them on the local label editor.	
Module label	Labels in which the I/O signals and buffer memory of a module to be used are already defined. By using the module labels, easy-to-use programs can be created without considering the internal address in the module. This label can be used in standard programs only.	Add module labels when adding the module information. Module labels are created as standard global labels.	Page 115 Registering Module Labels
System label	Labels that can be shared among iQ Works supported products. These labels are controlled by MELSOFT Navigator.	Register standard global labels as system labels on the global label editor.	Page 116 Registering System Label

For details on the label types, classes, and data types, refer to the following manuals.

📖 MELSEC iQ-R Programming Manual (Program Design)

📖 MELSEC iQ-F FX5 Programming Manual (Program Design)

5.2 Registering Labels

This section explains how to register the labels.

Configuration of label editor

This section explains the configuration of the label editor.
The editor to be displayed will differ depending on the label types.

Point

The display format and details of operation settings for each function can be set by setting the following option.
[Tool] ⇨ [Options] ⇨ "Other Editor" ⇨ "Label Editor Common"

Window

■Global label

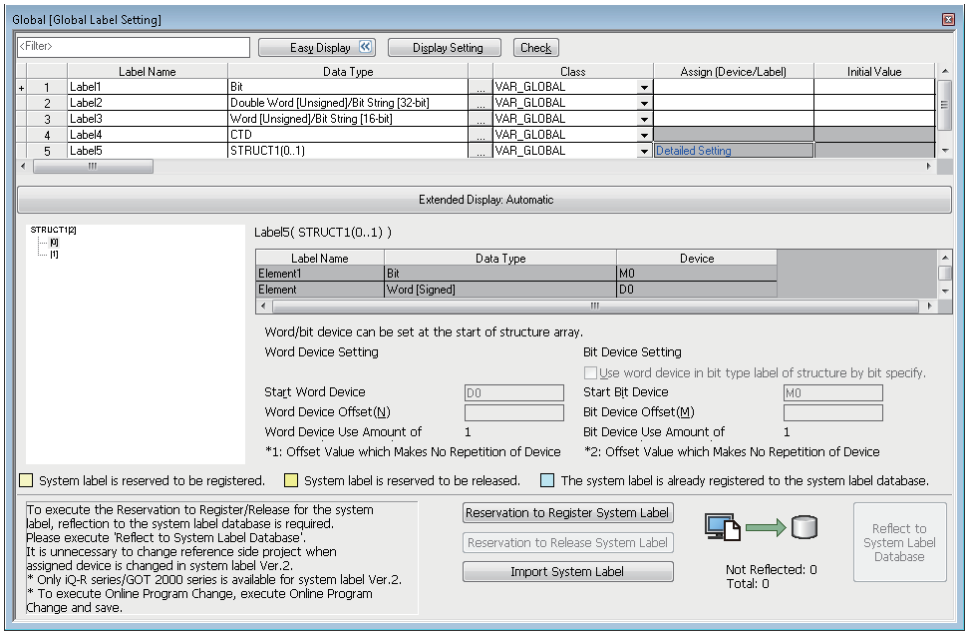
Project view ⇨ "Label" ⇨ "Global Label" ⇨ "(global label)"

■Local label

Project view ⇨ "Program" ⇨ "(execution type)" ⇨ "(program file)" ⇨ "(program block)" ⇨ "Local Label"

●Toolbar

●Label editor (global editor)


Global [Global Label Setting]
Filter
Easy Display
Display Setting
Check
Label Name
Data Type
Class
Assign (Device/Label)
Initial Value
1 Label1 Bit VAR_GLOBAL
2 Label2 Double Word [Unsigned]/Bit String [32-bit] VAR_GLOBAL
3 Label3 Word [Unsigned]/Bit String [16-bit] VAR_GLOBAL
4 Label4 CTD VAR_GLOBAL
5 Label5 STRUCT1(0..1) VAR_GLOBAL Detailed Setting
Extended Display: Automatic
Label5(STRUCT1(0..1))
Label Name
Data Type
Device
Element1 Bit M0
Element Word [Signed] D0
Word/bit device can be set at the start of structure array.
Word Device Setting
Bit Device Setting
Use word device in bit type label of structure by bit specify.
Start Word Device
D0
Start Bit Device
M0
Word Device Offset(N)
Bit Device Offset(M)
Word Device Use Amount of
1
Bit Device Use Amount of
1
*1: Offset Value which Makes No Repetition of Device
*2: Offset Value which Makes No Repetition of Device
System label is reserved to be registered.
System label is reserved to be released.
The system label is already registered to the system label database.
To execute the Reservation to Register/Release for the system label, reflection to the system label database is required.
Please execute 'Reflect to System Label Database'.
It is unnecessary to change reference side project when assigned device is changed in system label Ver.2.
* Only iQ-R series/GOT 2000 series is available for system label Ver.2.
* To execute Online Program Change, execute Online Program Change and save.
Reservation to Register System Label
Reservation to Release System Label
Import System Label
Not Reflected: 0
Total: 0
Reflect to System Label Database

- Click the [Display Setting] button to select the items to be displayed.
- Click the [Check] button to check errors before converting programs.
- When the data type is structure or function block, the corresponding hierarchy of labels is displayed.

Point

Font color, background color, and fonts can be changed.
☞ Page 48 Checking and Changing Colors and Fonts

Editing rows

■Adding rows

When using the New Declaration (After) function, the selected label is copied and added it to the row right after the specified row, with a value appended.

If a value is already appended after the label name, the data is copied with an incremented value.

When devices are set for global labels, the data is copied with an incremented device number.

For adding blank rows or setting increment regulation (decimal/hexadecimal), set the following option.

- [Tool] ⇨ [Options] ⇨ "Other Editor" ⇨ "Label Editor Common" ⇨ "Editor Setting"

Operating procedure

Select [Edit] ⇨ [New Declaration (Before)]( )/[New Declaration (After)]( ).

■Deleting rows (deleting labels)

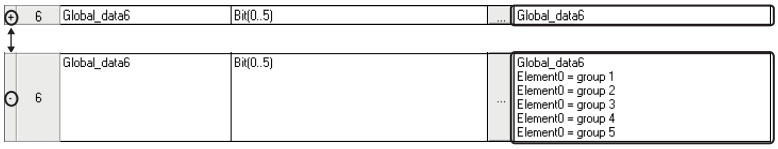
Operating procedure

Select [Edit] ⇨ [Delete Row]( ).

■Displaying all lines or first line of comments

For the "Comment" columns, data can be entered in multiple lines.

Double-click "+" or "-" to switch the display between all lines or only the first line.



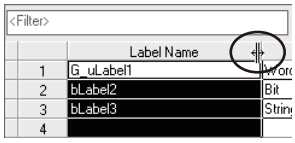
■Sorting labels

Labels can be sorted by clicking a title name of label editor.

For ascending order, is displayed, and as for descending order, is displayed in the title name.

■Filtering display

- If the wild card ('*', '?') is specified as a filtering condition, it is not applied as a filtering condition but displayed strings which are used '*' and '?'.
- To select the filtering target column, click the boundary on the right side of the column to as shown below and perform filtering.



- When filtering columns by selecting "Access from External Device", specify '1' after selecting "Access from External Device". As for it is not selected, specify '0'.

■Moving a row upward

Delete an empty row automatically, and move upward on the label editor.

Operating procedure

Select [Edit] ⇨ [Delete Blank Rows]. ( )

Entering information

Label name

Set label names except for the following names:

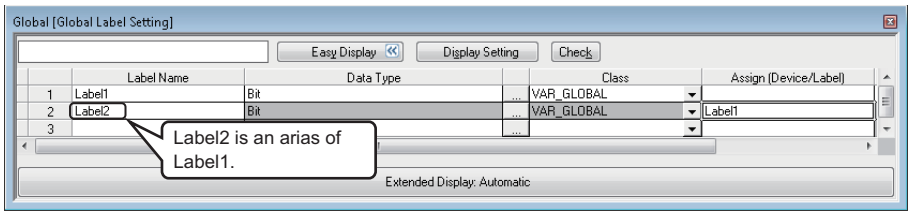
- A label name that includes a space.
- A label name that begins with a numeral.
- A label name which is same as the one used for devices.

Precautions

For label names, constants, and device initial values, use the characters in the Unicode Basic Multilingual Plane.
The label names and constants that specifies the characters other than the Unicode Basic Multilingual Plane cannot be used in programs.
The initial values specified with the characters outside the Unicode Basic Multilingual Plane may not be edited in the program.

■Alias

An alternative name for label that is assigned to a label is referred to as 'alias'.
For the data types, classes, initial values, and constants to which the alias function is set, the information of assigned devices/labels will be succeeded.



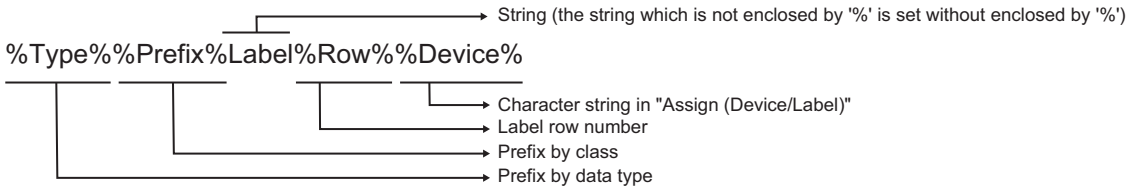
■Automatic naming rule

By selecting the row of "Data Type" or "Class" when the label name is blank , label name is set automatically.
Set the automatic naming rules in the following option setting.

- [Tool] ⇨ [Options] ⇨ "Other Editor" ⇨ "Label Editor Common" ⇨ "Editor Setting"

The automatic naming rules are as follows.

●Defined characters


String (the string which is not enclosed by '%' is set without enclosed by '%')
%Type%%Prefix%Label%Row%%Device%
Character string in "Assign (Device/Label)"
Label row number
Prefix by class
Prefix by data type

*The character other than the defined one enclosed by '%' will be blank.

The prefixes for each data type are as follows.

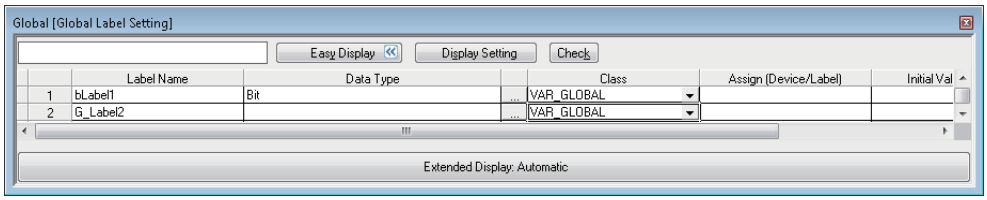
Data type	Prefix
Bit	b
Word [Unsigned]/Bit String [16-bit]	u
Double Word [Unsigned]/Bit String [32-bit]	ud
Word [Signed]	w
Double Word [Signed]	d
FLOAT [Single Precision]	e
FLOAT [Double Precision]	le
Time	tm
String	s
String [Unicode]	ws
Pointer	pd
Timer	td

Data type	Prefix
Counter	cd
Long Counter	lcd
Retentive Timer	std
Long Retentive Timer	lstd
Long Timer	ltd
Structure	st
FB	fb

The prefixes that are added depending on the selected class are as follows.

Class	Prefix
VAR_GLOBAL	G_
VAR_GLOBAL_RETAIN	GR_
VAR_GLOBAL_CONSTANT	GC_
VAR	Not added.
VAR_RETAIN	r_
VAR_CONSTANT	c_
VAR_INPUT	i_
VAR_OUTPUT	o_
VAR_IN_OUT	io_
VAR_OUTPUT_RETAIN	or_
VAR_PUBLIC	pb_
VAR_PUBLIC_RETAIN	pbr_

●Setting example



- When "Bit" is specified to data type: bLabel1
  Prefix 'b' that indicates the bit type + character string 'Label' + label row number '1'
- When "VAR_GLOBAL" is selected to the class: G_Label2
  Prefix 'G_' that indicates the class + character string 'Label' + label row number '2'

■Automatic synchronization

When label names/alias names are changed on the label editor, the changed label names/alias names used in a program editor or each editor will also be replaced automatically.

The labels are distinguished between global labels and local labels in the automatic synchronization. (☞ Page 120 Entering global labels/local labels)

The programs in which the replaced labels are used will be in the unconverted state.

If labels names/alias are changed, check if the change affects control programs using the Cross Reference function.

Data type

A data type can be selected on the "Data Type Selection" screen displayed by clicking [...] in the data type column on each label editor as well as the direct input.

There are three kinds of data types; "Simple Types", "Structured Data Type", and "Function Block". The data types that can be specified differ.

Array can be set for the specified data type.

For details on the data types, structures, and arrays, refer to the following manuals.

📖 MELSEC iQ-R Programming Manual (Program Design)

📖 MELSEC iQ-F FX5 Programming Manual (Program Design)

■Simple type

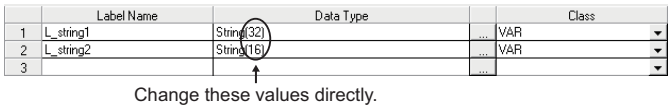
Specify a data type in "Data Type".

Labels of the following data types can be used for safety programs.

- Bit
- Word [Unsigned]/Bit String [16-bit]
- Double Word [Unsigned]/Bit String [32-bit]
- Word [Signed]
- Double Word [Signed]
- Time
- Timer
- Counter
- Retentive Timer

Point

Enter the data length of the string data type directly.



The initial value of the data length of a string data type can be set in the option setting.

- [Tool] ⇨ [Options] ⇨ "Other Editor" ⇨ "Label Editor Common" ⇨ "Data Type Setting"

■Structure

When specifying a structure data as a data type, creating a definition of the structure is required in advance. (☞ Page 114 Creating structure definition)

After creating a structure definition, specify the structure name in "Data Type".

■Function block

When specifying a function block data as a data type, creating a function block is required in advance.

For details, refer to the following section.

☞ Page 214 Creating Function Blocks

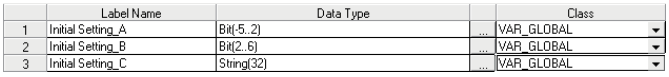
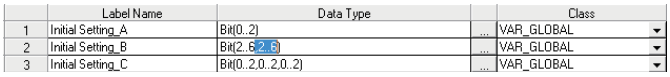
After creating a function block, specify the function block name in "Data Type".

■Setting arrays for data type

Select "ARRAY", and enter the number for "Element".

Set the data type of the array element in the same manner as setting the normal data type.

The method of how to edit the arrays on each editor is as follows.

Description	Operating procedure
Change the offset	To change the offset ([Array start value]..[Array end value]) to a value other than '0', edit the offset value by entering the array declaration directly in text. Minus values can be set for offset. 
Change the array to a two- or three-dimensional array	Enter the array declaration directly in text. For details, refer to the following manual. 📖 MELSEC iQ-R Programming Manual (Program Design) 

Class

Select the class from the pull-down list of "Class".

Safety global labels, standard/safety shared global labels, local labels of a safety program, and local labels of a safety FB do not support the latch type class (a class of which name includes 'RETAIN').

For details on the classes, refer to the following manuals.

📖 MELSEC iQ-R Programming Manual (Program Design)

📖 MELSEC iQ-F FX5 Programming Manual (Program Design)

Assignment (devices/labels)

Assign the arbitrary devices/labels to the global labels. (Undefined labels can also be assigned.)

The devices/labels which can be assigned differ depending on the assignment target label.

○: Can be assigned, ×: Cannot be assigned

Global label to be assigned	Assignment target				
	Standard device	Standard global label	Safety device	Safety global label	Standard/safety shared global label
Standard global label	○	○	×	×	×
Safety global label	×	×	○	○	×
Standard/safety shared global label	×	×	×	×	○

A digit-specified bit device (K4M0) and digit-specified word device (D0.1) can be specified. The module labels can also be assigned as a label.

If do not assign a device/label to the label, a label memory is assigned instead. For details on the label memory, refer to the following manuals.

📖 MELSEC iQ-R CPU Module User's Manual (Application)

📖 MELSEC iQ-F FX5 User's Manual (Application)

Precautions

- When specifying a timer or counter device to "Assign (Device/Label)", the assigned device is regarded as a contact (TS, STS, CS) if a bit type device is specified to "Data Type".
  When "Data Type" is a word type, the specified device is regarded as a current value (TN, STN, CN).
- When specifying the step relay(S) which does not specify a block No. (BL□) in "Assign (Device/Label)", the assigned label operates the same as the step relay(S) which does not specify a block No. (BL□). Therefore, the same labels operate differently depending on the location of use. For details on the step relay(S) , refer to the following manual.

📖 MELSEC iQ-R Programming Manual (Program Design)

■When data type is structure

A device can be assigned to the structure element on the Structure Device Setting Window displayed when clicking the "Detailed Setting" button. (The color of "Detailed Setting" is displayed in pink when no setting exists, and is displayed in blue when the items have been set.)

When the structure array is specified, device numbers can be set in a fixed interval for each element of the structure array by specifying offset. When '0' is specified for the offset, the same device number set to the start device data is set to all data in the array.

Initial value

Set an initial value to the label.

FX5CPU does not support it.

However, if a device is assigned to a global label, the label does not operate with the initial value of the label.

The availability of initial values differs depending on the data types and classes.

The entry method of an initial values is the same as that of the constants of labels. For details, refer to the following manual.

📖 MELSEC iQ-R Programming Manual (Program Design)

In addition, the initial values of safety global labels, local labels of a safety program, and standard/safety shared global labels cannot be set.

■Applicable range

Initial values of labels will be set when the CPU module is at STOP to RUN. If the values of labels are changed by program, the program operates with the changed initial values.

■Initial values of label of which data type is array

Different initial values for the respective array elements cannot be set. When setting the different values for each array, set the initial value by program.

Constant

Constant can be set when "VAR_GLOBAL_CONSTANT" or "VAR_CONSTANT" is specified to the class.

The availability of the constants differs depending on the data type and class.

For details on how to enter the constants, refer to the following manuals.

📖 MELSEC iQ-R Programming Manual (Program Design)

📖 MELSEC iQ-F FX5 Programming Manual (Program Design)

Comment

Append a comment to the defined labels.

When entering comments on the label editor, press the Ctrl + Enter keys to insert a line feed (two characters are used) in a cell.

Two or more comments can be set for one label. Set the display settings of the multiple comments on the "Multiple Comments Display Setting" screen.

For details, refer to the following section.

☞ Page 49 Setting Comment Display

■Copying device comments

The comment of the device assigned to the selected label can be copied to the label editor.

Operating procedure

Select [Edit] ⇨ [Copy Device Comment] ( ).

Accessing from external devices

Select this to monitor devices/labels from external devices connected to the CPU module.

FX5CPU does not support it.

If the selection status of "Access from External Device" is changed, the change will not be reflected to the assigned devices/labels.

An error occurs when accessing from an external device to the CPU module in any of the following condition.

- Data type is pointer type/FB
- Class is "VAR_GLOBAL_CONSTANT"
- Indexed devices

This setting is not available for safety global labels nor standard/safety shared global labels.

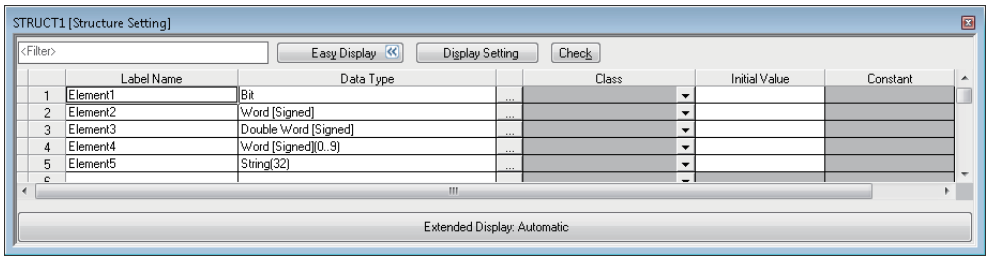
Creating structure definition

Set the structure elements.

Be sure to create the data for structure definition in advance. (☞ Page 72 Creating projects)

Window

Project view ⇨ "Label" ⇨ "Structured Data Types" ⇨ "(structure)"



- Click the [Display Setting] button to select the items to be displayed.
- Click the [Check] button to check errors before converting programs.

Importing/exporting files

Import/export the data of the label editor to/from a file.

Operating procedure

1. Open the label editor.
2. Select [Edit] ⇨ [Import File] ( )/[Export to File] ( ).

Formatting CSV files

The header titles (Class, Label Name, Data Type, etc.) of a label editor and a CSV file are linked.

- The data of which header title matches with that of the CSV file is imported to the label editor. When excluding the specific data, delete the column from the CSV file.
- When importing the CSV file exported from GX Works3 with different language, edit the header title name of the CSV file to match to the header title displayed on the label editor of the import target GX Works3.
- The data can be imported even when the column order in CSV file does not match with that in the label editor.
- When "Access from External Device" is selected, '1' is output. As for it is not selected, '0' is output. When editing a CSV file, set '1' or '0'.

Point

The files exported from GX Works2 can be imported to GX Works3.

Before importing the CSV file exported from GX Works2, edit the header title name of the CSV file to match the header title on the label editor of GX Works3.

5.3 Registering Module Labels

This section explains how to register the module labels.

Registering to global label

Module labels are registered by clicking the [Yes] button on the confirmation screen displayed after performing either of the following operations.

- Place a module on the Module Configuration Diagram and fix the parameters.
- Add a new module from the Project view.

The registered module labels are displayed in "Global Label" in the Project view and in the [Module] tab of the Element Selection window.

When using a module label in a program, enter a module label name directly or drag and drop it from the Element Selection window.

Point

To use a module label under arbitrary name, set alias to the module label.

5

Precautions

■Considerations when registering module labels

Since all the registered module labels are written to a CPU module, the memory capacity of the CPU module may be exceeded. In this case, perform any of the following operations.

- Delete the module labels that are not used. (☞ Page 108 Deleting rows (deleting labels))
- Change the write target of the global labels to SD memory card.
- For FX5CPU, the writable capacity is the same as that of the CPU module even when data is written to an SD memory card.

■Module label with the name that starts with 'zReserve'

Do not write data to the module label with the label name that starts with 'zReserve' among the module labels. Doing so may cause malfunction of the programmable controller systems.

Ex.

'Instance name'_'Module number'.'Label name' GF11_1.zReserveAreaSB00007

Editing module labels

The editing on the label editor of the module label (M+Global) which has been registered into the global label, only the deletion with selecting the row is applicable.

Do not edit or add labels on the label editor.

Re-registering deleted module labels

When registering the module labels which are once deleted from the global label editor, select a module name displayed in the [Module] tab of the Element Selection window and right-click it and select [Add Module Label] from the shortcut menu.

Editing structures

When module labels are registered, the definition of the structure in the module labels are registered in "Structured Data Types" of the Project view.

A defined structure can be copied on the Project view. The copied defined structure name of the module label is changed from "+" to "_".

5.4 Registering System Label

This section explains how to register the global labels as the system labels.
Only standard global labels support system labels.

System label

Using the system labels controlled by MELSOFT Navigator enables programming with the same label names among the iQ Works supported products (GX Works3, MT Developer2, GT Designer3).
Once a workspace is saved with MELSOFT Navigator, a database to manage system labels is created in the workspace.
System labels cannot be used in a workspace without the system label database.

Precautions

Since module labels are read only, they cannot be registered as system label.

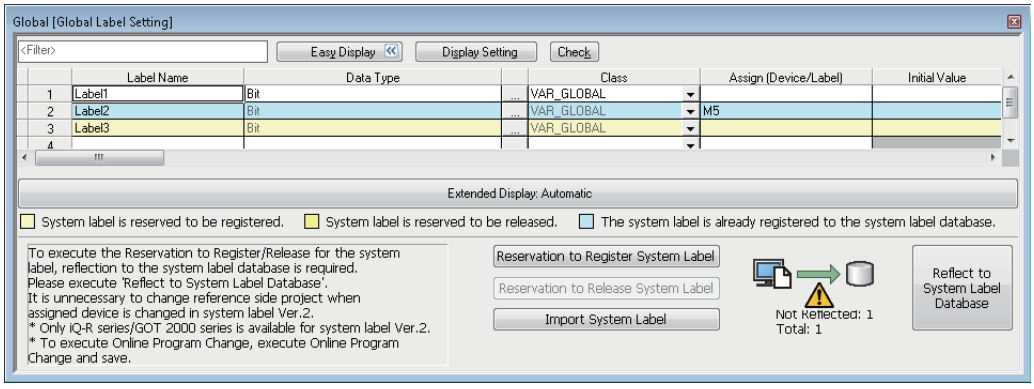
System label version

There are two types of system labels for MELSOFT Navigator: system label ver.1 and system label ver.2.
FX5CPU only supports system label ver.1.
For more details on the differences between system label ver.1 and system label ver.2, and creation methods for each system label database, refer to the Help of MELSOFT Navigator.

Configuration of label editor

Window

Project view ⇨ "Label" ⇨ "Global Label" ⇨ "(global label)"


Global [Global Label Setting]
<Filter>
Easy Display
Display Setting
Check
Label Name
Data Type
Class
Assign (Device/Label)
Initial Value
1 Label1 Bit VAR_GLOBAL
2 Label2 Bit VAR_GLOBAL M5
3 Label3 Bit VAR_GLOBAL
4
Extended Display: Automatic
System label is reserved to be registered.
System label is reserved to be released.
The system label is already registered to the system label database.
To execute the Reservation to Register/Release for the system label, reflection to the system label database is required.
Please execute 'Reflect to System Label Database'.
It is unnecessary to change reference side project when assigned device is changed in system label Ver.2.
* Only iQ-R series/GOT 2000 series is available for system label Ver.2.
* To execute Online Program Change, execute Online Program Change and save.
Reservation to Register System Label
Reservation to Release System Label
Import System Label
Not Reflected: 1
Total: 1
Reflect to System Label Database

Displayed items

Item	Description
System Label Relation	Displays the relation between the global label and the system label. • Disclose: The status in which the global labels are open as the system labels • Reference: The status in which the system labels opened by other projects are imported in the global labels • Blank: The status in which the global labels is not related to the system labels
System Label Name	Displays the system label name related to the global labels.
Attribute	Displays the attribute of the system label which is related to the global labels.

Registering labels in system label database

Disclose standard global labels created with GX Works3 as system labels.
When registering in the system label database at the first time, create a workspace with MELSOFT Navigator and save projects of GX Works3 in the workspace in advance.

Operating procedure

1. Display the global label editor.
2. Select the labels to be registered as system label, and click the [Reservation to Register System Label] button.
3. Click the [Reflect to System Label Database] button.

The "Check before registering in system label data base" screen is displayed.

4. Check the list of labels to be registered, and click the [Register] button.

Precautions

The structure array type global labels to which "Structure Array Offset Value" is set cannot be registered as system labels.

Importing system labels in system label database to GX Works3

Import the system labels registered with MELSOFT Navigator or other projects to a project.

Operating procedure

1. Display the global label editor.
2. Click the [Import System Label] button.

The "Import System Labels to Project" screen is displayed.

3. Select the system labels to be imported, and click the [Import] button.

Releasing system labels

Release the relation with the system label database, and restore system labels to normal global labels.

Operating procedure

1. Display the global label editor.
2. Select the labels of which relation is to be released from the labels registered as system labels.
3. Click the [Reservation to Release System Label] button.
4. Click the [Reflect to System Label Database] button.

The "Check before registering in system label data base" screen is displayed.

5. Check the list of labels to be released, and click the [Register] button.

Verifying system label information

Compare the system label information of GX Works3 projects to the system label database, and check if there is any difference.
If differences exist, correct the system label information of the GX Works3 projects, and synchronize the information with the system label database.

Operating procedure

1. Select [Edit] ⇨ [System Label] ⇨ [Execute Verification Synchronous with System Label].
2. Check and correct information on the "Execute Verify and Synchronize with system label" screen.
3. Click the [Reflection] button.

Importing the changes of system label database

Import the changed contents to the GX Works3 projects when the system labels used in a project have been changed in other projects.

Importing the changes automatically

Operating procedure

Click the [Yes] button on the message to confirm whether or not import the changed contents, which appears when any of the following operations is performed in a state where a system label change notification has been received.

- Opening a project
- Saving a project
- Reflecting labels to system label database
- Displaying the "Online Data Operation" screen
- Starting the Simulation function

Click the [No] button to import them after checking the changed contents. (☞ Page 118 Importing after checking the changes)

Importing after checking the changes

When the system label information in the GX Works3 projects is different from the system label database, the icon to notify the changes is displayed at the lower-left corner of the GX Works3 screen.
Check the contents that have been changed when the icon is displayed. In addition, import the changed contents to the GX Works3 projects.

Operating procedure

1. Select [Edit] ⇨ [System Label] ⇨ [Check the changes of System Label Database].
2. Check the information on the "Import Change Contents of System Label Database" screen.
3. Click the [Import] button.

PART 3 PROGRAMMING

This part explains the functions of the editor to edit a sequence program and the settings of device memory/ device initial values.

6 CREATING PROGRAMS

7 SETTING DEVICE MEMORY

8 SETTING DEVICE INITIAL VALUES

9 SEARCHING DATA

10 SEGMENTING PROGRAMS

6 CREATING PROGRAMS

This chapter explains how to create a program.

6.1 Programming Functions

Features and types of programming language

The programming languages supported by GX Works3 are as follows:

Program language	Name	Description	Creation method of program	Language specification
Ladder*1	Ladder Diagram	A graphic language using ladder programs composed of contacts and coils. The inline structured text functions to edit ST programs on the ladder editor can be used.	Page 123 Creating Ladder Programs	MELSEC iQ-R Programming Manual (Program Design)
ST	Structured Text	Control syntax such as selection branch by conditional syntax or repetitions by iterative syntax can be controlled, as in the high-level language such as C language. By using these syntax, concise programs can be written.	Page 145 Creating ST Programs	
FBD/LD	Function Block Diagram/Ladder Diagram	A graphic language for creating a control program only by placing and connecting an element.	Page 151 Creating FBD/LD Programs	
SFC	Sequential Function Chart	A graphic language for clarifying execution orders and execution conditions.	Page 158 Creating SFC Programs	

*1 Safety programs are supported.

Features and types of POU

The POUs which can be used in a program are as follows.

POU	Description	Reference
Function block	Outputs an operation result in accordance with values of the internal memory.	Page 214 Creating Function Blocks
Function	Always outputs the same operation result against the same input value.	Page 217 Creating Functions

Inputting global devices/local devices

To use global devices and local devices individually, enter the devices as follows:

Note that FX5CPU does not support local devices.

- Standard global device: Enter a device name (Example: D10)
  Safety global device: Prefix 'SA\' to the device name (Example: SA\D10)
- Standard local device: Prefix '#' to the device name (Example: #D10)
  Safety local device: Prefix 'SA\#' to the device name (Example: SA\#D10)

For details of the global device/local device, refer to the following manual.

MELSEC iQ-R CPU Module User's Manual (Application)

Entering global labels/local labels

When a label with the same name exists in both global labels and local labels, the label in a program is handled as follows:

- After a global label and a local label are registered, the label names are entered in the program: Treated as a local label.
- After a global label is registered, the label name is entered in the program, and then a local label with the same name is registered: Treated as a global label.
- After a local label is registered, the label name is entered in the program, and then a global label with the same name is registered: Treated as a local label.

The global labels and local labels can be distinguished by setting the different color on the "Color and Font" screen.

Creation procedure

1. Create data of program block. (☞ Page 72 Creating projects)
2. Set an execution order and an execution type. (☞ Page 121 Program execution order/execution type settings)
3. Create a program. (☞ Page 123 Creating Ladder Programs, Page 145 Creating ST Programs, Page 151 Creating FBD/LD Programs, Page 158 Creating SFC Programs)
4. Check the created program. (☞ Page 183 Checking Programs)
5. Convert the created program. (☞ Page 184 Converting Programs)

6.2 Program execution order/execution type settings

This section explains how to set an execution order and an execution type for a program.

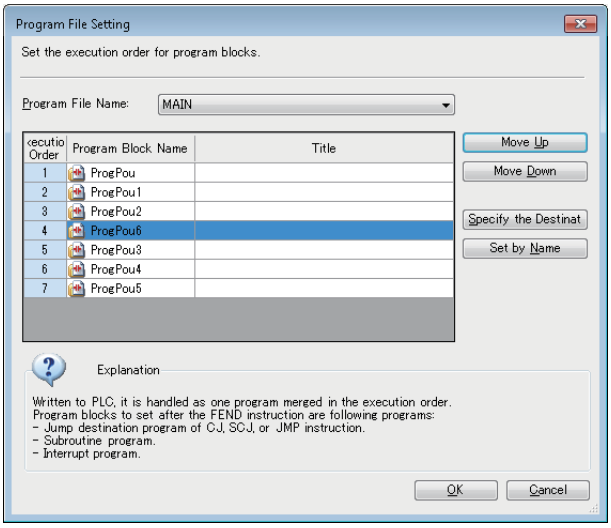
Program execution order settings

Set an execution order of program block in a program file.

An SFC program is executed according to the block No. Check the block No. on an SFC block list.

Window

- [Convert] ⇨ [Program File Setting]
- Select a program file on the Project view, then right-click and select [Program File Setting] from the shortcut menu.



Operating procedure

1. Select a program file name.
2. Set an order to be executed to a program block, then click the [OK] button.

Point

A program file can be sorted in the execution order specified in the program file settings. Select a program file on the Project view, then right-click and select [Sort] ⇨ [Execution Order] from the shortcut menu.

Program execution types

One of the following execution types can be specified to a program: Initial, Scan, Fixed Scan, Event, Standby, or No Execution Type.

For details, refer to the following manuals.

📖 MELSEC iQ-R CPU Module User's Manual (Application)

📖 MELSEC iQ-F FX5 User's Manual (Application)

Only the Fixed Scan can be specified in safety programs.

Setting method

To set the execution type, right-click the program name of which execution type is to be set, and select [Register Program] from the shortcut menu or drag and drop it on the Project view.

The execution type set on the Project view is applied to "Program" of the CPU parameter.

6.3 Creating Ladder Programs

This section explains the method for creating a ladder program.
The details on the specification of ladder program are described in the following manual. Please read it in advance.
📖 MELSEC iQ-R Programming Manual (Program Design)

Point

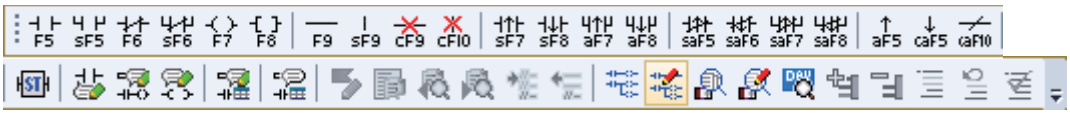
The display format and detailed operations for each function can be set by the following option setting.
[Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "Ladder Editor"

Configuration of ladder editor

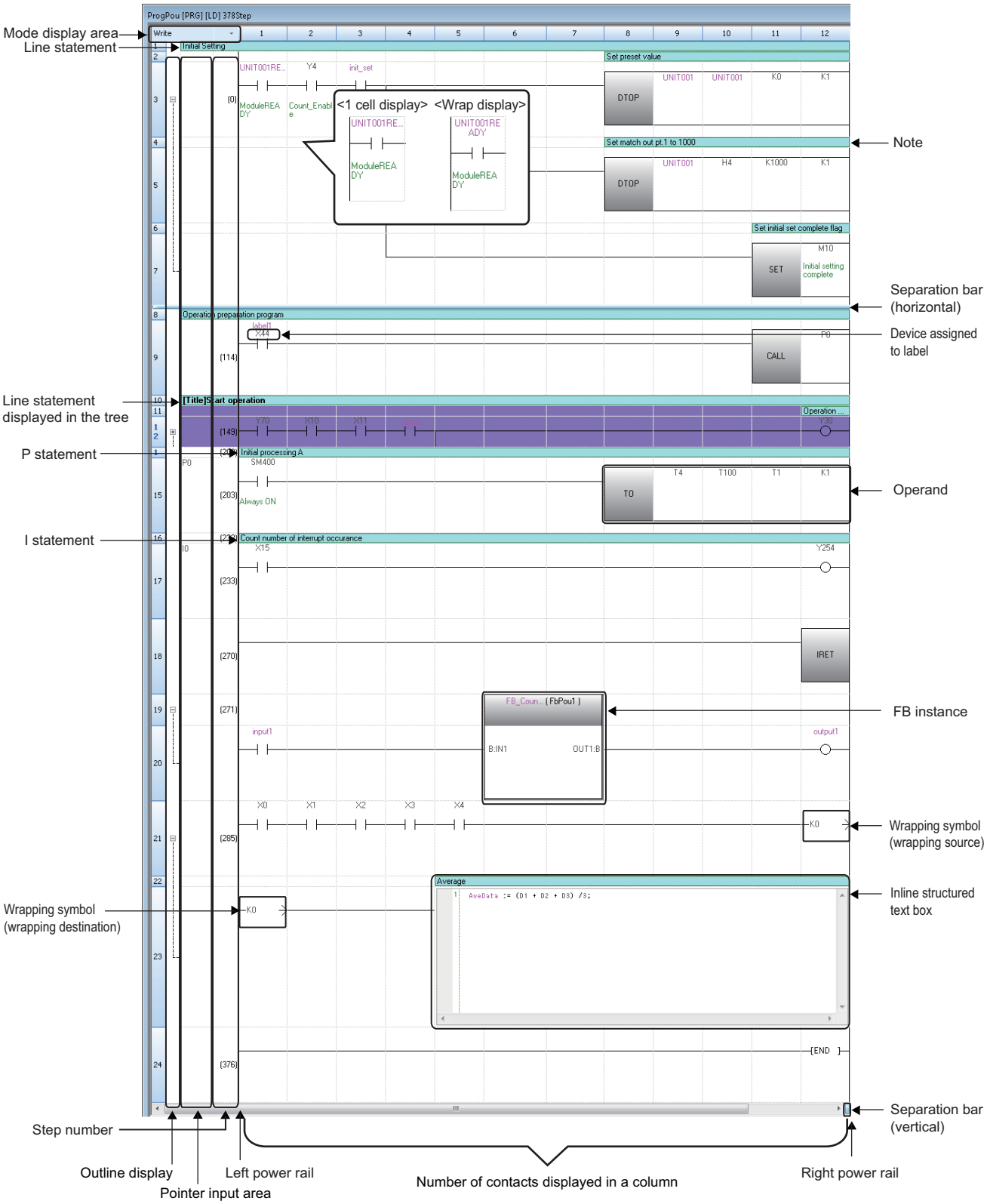
Window

Project view ⇨ "Program" ⇨ "(execution type)" ⇨ "(program file)" ⇨ "(program block)" ⇨ "Program"

●Toolbar



●Ladder editor



Displayed items

Item	Description	Related operation
Mode display area	An area that displays mode such as Write/Read/Write Mntr/Read Mntr.	■Switching mode ☞ Page 126 Read mode/Write mode/Monitor read mode/Monitor write mode
Step No.	The start Step No. of a ladder block.	—
Device assigned to label	A device assigned to a label. Editing and searching/replacing labels displayed on a ladder editor are available. A device with any of the following conditions is not displayed. • A device that is assigned by converting the program • When a label is used for array elements • When using a structure label	■Settings ☞ Page 107 Registering Labels ■Show/Hide [View] ⇨ [Display Device] ( )
Alias	An alternative name for a label. Editing and searching/replacing can be performed for the labels displayed on the ladder editor.	
Comment	A comment set to a device/label.	■Entering elements ☞ Page 135 Entering comments/statements/notes ■Show/Hide [View] ⇨ [Comment Display]/[Statement Display]/[Note Display]
Line statement	A comment that is appended to a ladder block.	
Line statement displayed in the tree	A line statement that is displayed in the tree of the Project view.	
P statement	A comment that is appended to a pointer number.	
I statement	A comment that is appended to an interrupt pointer number.	
Note	A comment that is appended to a coil/application instruction in the program.	
Inline structured text box	An area that can edit ST programs on the ladder editor.	■Insertion ☞ Page 134 Inserting inline structured text
FB instance	An entity of a function block used in a ladder program.	■Insertion ☞ Page 128 Inserting function blocks, Page 133 Inserting functions
Operand	A value or label name/device name to be operated.	—
Outline display	A symbol that indicates the status (collapse/expand) of a ladder block.	■Show/Hide • [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "Ladder Editor" ⇨ "Ladder Diagram" ⇨ "Display Format" • [View] ⇨ [Outline] ⇨ [Show/Hide of Outlines]
1 cell display	The rest of a device/label name is omitted by ellipses "..." in order for the name to be fit in one cell.	■Switching display [View] ⇨ [Change Display Format of Device/Label Name] ⇨ [1 Cell Display]/[Wrapping Ladder Display]
Wrap display	A long device/label name is wrapped to be fit in one cell. If the label name still cannot be displayed in one cell, the rest of the name is omitted by ellipsis "...".	
Left power rail	The power rails of a ladder program.	—
Right power rail		
Tool hint	The information on device/label where the mouse cursor is placed over is displayed.	■Changing display contents [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "Ladder Editor" ⇨ "Tool Hint"
Number of contacts displayed in a column	The maximum number of cells which are occupied with contacts, coils, and instructions. When the maximum number of contacts is exceeded, the line will be wrapped automatically.	■Changing number of contacts [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "Ladder Editor" ⇨ "Ladder Diagram" ⇨ "Display Format"
Wrapping symbols	The wrapping symbols are displayed at wrapping source and wrapping destination. The same numbers (serial numbers) are assigned to the wrapping source symbol and the wrapping destination symbol that are created in pairs. Wrapping lines cannot be created on the input and output circuits of an FB instance.	—

Point

A font color, background color, and fonts can be changed.

☞ Page 48 Checking and Changing Colors and Fonts

Read mode/Write mode/Monitor read mode/Monitor write mode

The program cannot be edited directly when the mode is switched to the Read mode/Monitor read mode.
However, performing any of the following operations changes the programs.

- When the source function block name is changed or the function block is deleted
- When input/output labels of the source function block are changed and converted
- When the devices/labels are replaced with the replacement function by specifying the whole project

To prohibit editing the program, use the security function.

Point

- The mode can be changed from the pull-down list on the mode display area.
- When "No" is selected in the following option, the ladder editor can be edited directly same as Write mode in Read mode.
  [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "Ladder Editor" ⇨ "Ladder Diagram" ⇨ "Display Format" ⇨ "Use the Switching Ladder Edit Mode (Read, Write, Monitor, Monitor (Write))"

Entering ladder programs

This section explains how to enter a ladder program.

For details of ladder symbols, refer to the following manuals.

📖 MELSEC iQ-R Programming Manual (Program Design)

📖 MELSEC iQ-F FX5 Programming Manual (Program Design)

Inserting contacts, coils, instructions, arguments

Operating procedure

■Inserting from the element entry dialog

Enter an instruction and arguments on the element entry dialog[*2] displayed by selecting a cell[*1] to be entered or edited, and double-clicking or pressing the Enter key.

*1 For the "Insert mode" (press the Insert key to switch to the Overwrite mode), instructions are inserted on the cursor position. Therefore, if an instruction is already inserted in the selected cell, the existing instruction will be moved back.

*2 The cursor position on the ladder editor can be moved by pressing the Ctrl + ←/→/↑/↓ keys.
To select an instruction/argument, press the Alt+←/→ keys.
Options of instructions/labels can be displayed by pressing the ◆ Ctrl+Space keys.

■Inserting from the "Enter Ladder" screen with description

Select an instruction[*1] and enter arguments on the "Enter Ladder" screen displayed by clicking the [Extd Dspl] button on the element entry dialog.

*1 The frequently used instructions can be registered as "Favorites" by right-clicking and selecting [Add to Favorites] from the shortcut menu. The instructions added to "Favorites" can be selected in the lowest cell of "List" in the pull-down list.

■Inserting from the menu/toolbar

Select a cell to where an element is added on the ladder editor, then select [Edit] ⇨ [Ladder Symbol] ⇨ [(element)]. It can also be added by selecting from the toolbar.

■Inserting from the Element Selection window

Select a contact, coil, or instruction displayed in the Element Selection window, and drag and drop it on the ladder editor. After that, edit the arguments.

■Editing arguments directly

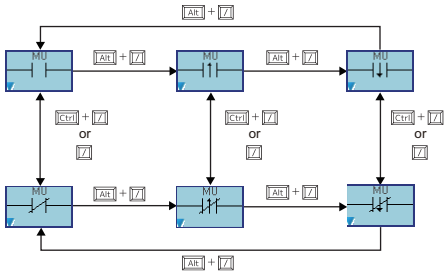
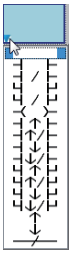
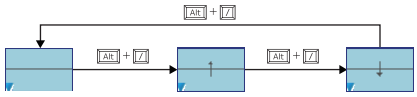
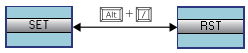
Select a cell where an argument is inserted, and press the F2 key to edit the argument.

Point

- The labels to be input as an argument can also be input using aliases.
- A device number/label name can be incremented (+1)/decremented (-1) by pressing the Alt+↑/↓ keys.
- When inserting an instruction without entering the arguments, '?' is added automatically. ('?' can be entered as an argument.)
  The arguments can be entered later.

■Switching methods for contacts/instructions

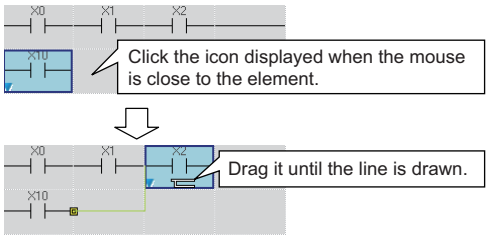
Select a contact/instruction to be switched, and select [Edit] ⇨ [Easy Edit] ⇨ [Switch Ladder Symbol Invert]/[Switch Pulse/Switch SET and RST Instructions] or follow the methods shown below.

Switching contacts and instructions	Shortcut key	Pull-down list
Switching open/close contact, raising/falling pulse		Select a symbol from the pull-down list displayed when clicking the blue triangle at the bottom left in the cell. 
Switching operation result rising/falling pulse		
Switching SET/RST instructions		—

Drawing lines

Draw a line in the program.

If a line has already been drawn, the line will be deleted.

Operation	Draw line
Drag and drop	
Keyboard	Ctrl + ←/→/↑/↓ keys
	Press the Ctrl + Shift + ←/→ keys to enter horizontal lines consecutively from a cursor position to the next contact/coil/line connection point.

Inserting function blocks

Paste a function block as a part, and then name (FB instance name) and insert it into a sequence program.

For the creation methods of a program of a function block, refer to the following section.

☞ Page 214 Creating Function Blocks

Operating procedure

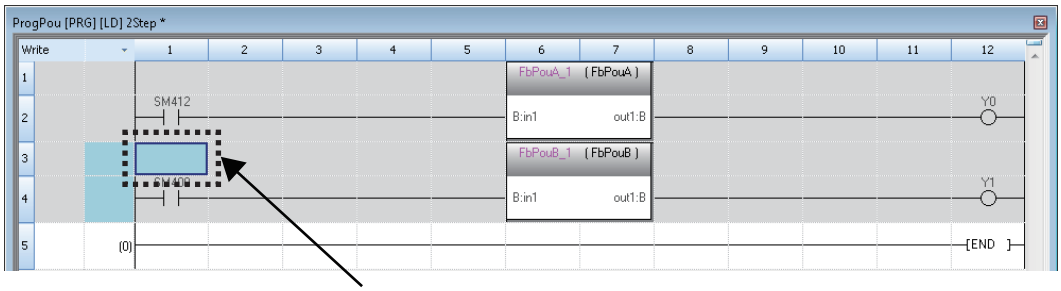
1. Drag and drop a function block element to an arbitrary cell on the sequence program from the Project view or Element Selection window.
2. Select which to register the function block either as a global label or as a local label in the list of "FB Instance Name" screen.

When the function block is pasted, the FB instance name is automatically registered as a label on the selected label setting screen.

3. Connect the input and output to the FB instance.

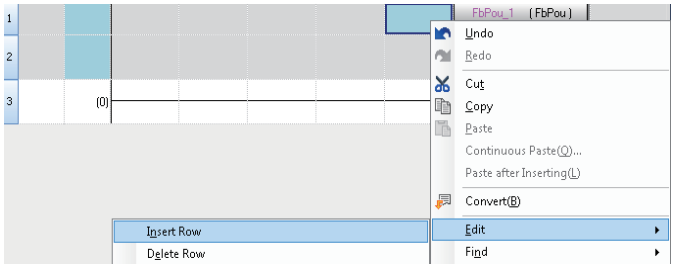
Point

- By entering 'fb.' in the element entry dialog, the options of function block elements are displayed. A function block element can be entered by selecting it from the options.
- To insert a function block between two consecutive FB instances, select [Edit] ⇨ [Insert Row] above the second FB instance first, and then insert the function block on the inserted row.



Place the cursor in the row on which the second FB instance exists, and insert a row.

- To insert a row above an FB instance/function, select a cell on the row with the definition name as shown below, and select [Edit] ⇨ [Insert Row].



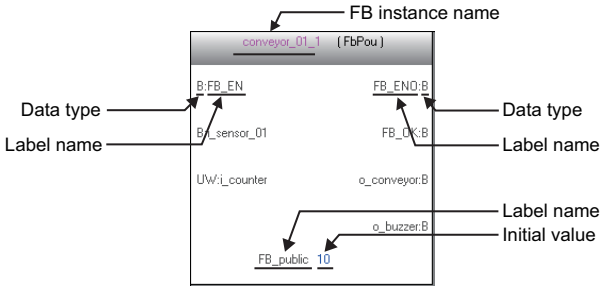
- To insert a row when either of the input argument or output argument for FB instance/function is other than BOOL data type, convert the argument to BOOL data type and insert a row, and then return the data type to the one other than BOOL.

Precautions

Pasting a function block to a sequence program causes the file size to increase.
Pasting the same function blocks to one sequence program repeatedly also causes the size increase of the sequence program in accordance with the number of pasted function blocks.

■Displaying FB instance

On the pasted FB instance, label names of I/O variables and their corresponding data types are displayed.



The following are the representation of data types.

- B : Bit
- W : Word
- D : Double word [signed]
- E : FLOAT (Single Precision)
- L : FLOAT (Double Precision)
- S : String

The initial values set on the label editor when creating FB program are also displayed. The initial values set to each FB instance are not displayed.

■Editing FB instance names

Operating procedure

1. Move the cursor on an FB instance.
2. Select [Edit] ⇨ [Edit FB Instance].
3. Enter a new FB instance name.

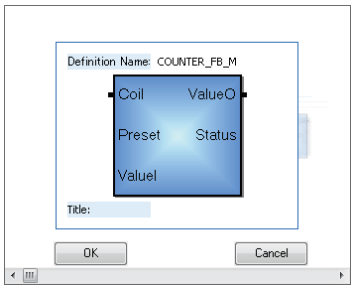
Point

- The name can be changed directly by moving the cursor on the FB instance and pressing the F2 key.
- The data of the function block can be edited directly by moving the cursor on the FB instance and pressing the / key.

■Replacing FB instances

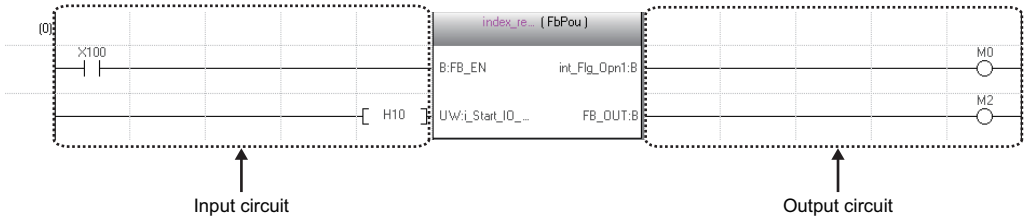
Operating procedure

1. Move the cursor on an FB instance.
2. Select [Edit] ⇨ [Change FB/FUN Data].
3. Scroll the displayed screen and select the function block to be replaced, and click the [OK] button.



■Creating input and output circuits of FB instances

Create input and output circuits of the FB instance pasted to a sequence program.



The number of contacts occupied by input circuits/output circuits of an FB instance differs depending on the setting of the following option.

- [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "Ladder Editor" ⇨ "Ladder Diagram" ⇨ "Display Format" ⇨ "Display Connection of Ladder Diagram"

Setting of the number of displayed contacts	Number of contacts in input circuit	Number of contacts in output circuit
9 contacts	4 contacts	3 contacts + 1 coil
11 contacts	5 contacts	4 contacts + 1 coil
13 contacts	6 contacts	5 contacts + 1 coil
17 contacts	8 contacts	7 contacts + 1 coil
21 contacts	10 contacts	9 contacts + 1 coil
33 contacts	16 contacts	15 contacts + 1 coil
45 contacts	22 contacts	32 contacts + 1 coil

Instructions that exceed the number of occupied contacts cannot be entered.
When instructions cannot be placed by changing the number of displayed contacts, the ladder block may not be displayed properly.

Operating procedure

1. Select [Convert] ⇨ [Convert] ( ).

The ladder block is converted, and lines are connected to the input and output labels of the FB instance.

2. Edit the input circuits.

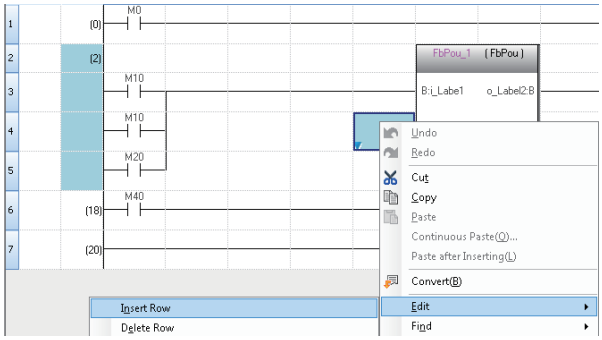
Enter an element to the input circuit in the same manner as creating normal ladder programs. Create a program in accordance with the data type of input variable.

3. Enter elements to the output circuit in the same manner as the input circuit.

Edit the program other than function blocks as well. After complete editing, convert the program.

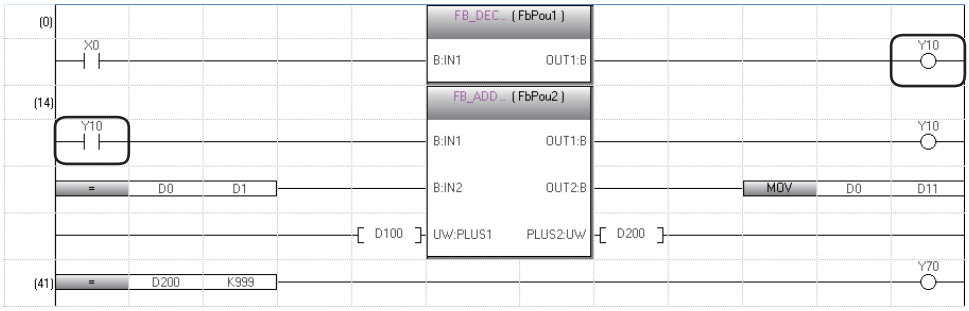
Point

A parallel circuit can be created in a input circuit on a FB instance.
To add an element between a parallel circuits, select a cell under the second row in the input circuit as shown below, and select [Edit] ⇨ [Insert Row] to enter an element in the added row.



■Considerations for using function block

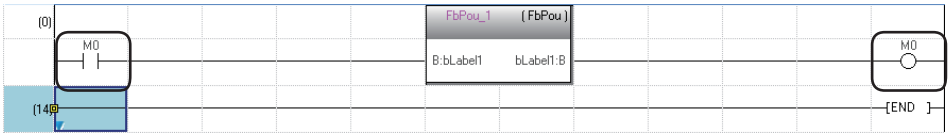
- A single function block can be pasted to a single ladder block.
  The output of an FB instance cannot be directly connected to the input of another FB instance.
  When connecting function blocks each other, use a coil to initially receive the output of a single function block, and then connect the contact of the coil to the input of the other function block.



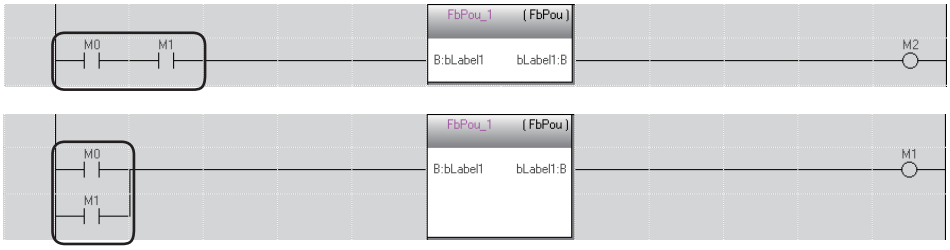
- When the label settings of a function block is changed, convert the program or all programs.

■Considerations for using function block that the class of label is "VAR_IN_OUT"

- If the version of GX Works3 is 1.011M or earlier, use the same device/label to both the input and output which are connected to the label of 'VAR_IN_OUT' ,otherwise, the program does not execute properly.



- The subroutine type function blocks that the devices/labels for the input and the output connecting to to the label of 'VAR_IN_OUT' are not same, the verification results are mismatched. It is due to the difference of the version of GX Works3 which was used for converting all programs (Version 1.011M of earlier, and Version 1.015R or later). In that case, convert all programs using the same version of GX Works3.
- For the version of GX Works3 is 1.015R or later, the macro/subroutine type function blocks will be an error when the multiple coils exist on the left side of "VAR_IN_OUT" in a program.



Inserting functions

Insert a function in a ladder program.

For the creation methods of functions, refer to the following section.

☞ Page 217 Creating Functions

Operating procedure

1. Drag and drop a function element to an arbitrary cell on the sequence program from the Project view or Element Selection window.
2. Connect an input and an output to the function.

Point

- By entering 'fun.' in the element entry dialog, the options of function elements are displayed. A function element can be entered by selecting it from the options.
- To insert a row after a function was inserted, refer to the point in "Inserting function blocks". (☞ Page 128 Inserting function blocks)

■Adding/deleting arguments

Only for functions of which number of arguments can be changed, the arguments can be added/deleted.

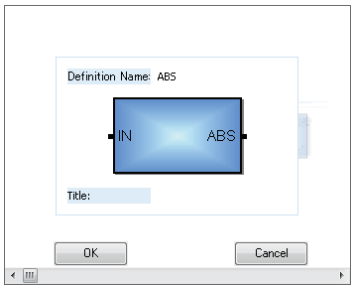
Operating procedure

1. Move the cursor on a function.
2. Select [Edit] ⇨ [I/O Argument] ⇨ [Increment Argument] ( )/[Delete Argument] ( ).

■Replacing functions

Operating procedure

1. Move the cursor on a function.
2. Select [Edit] ⇨ [Change FB/FUN Data].
3. Scroll the screen and select the function block to be replaced, and click the [OK] button.



Inserting inline structured text

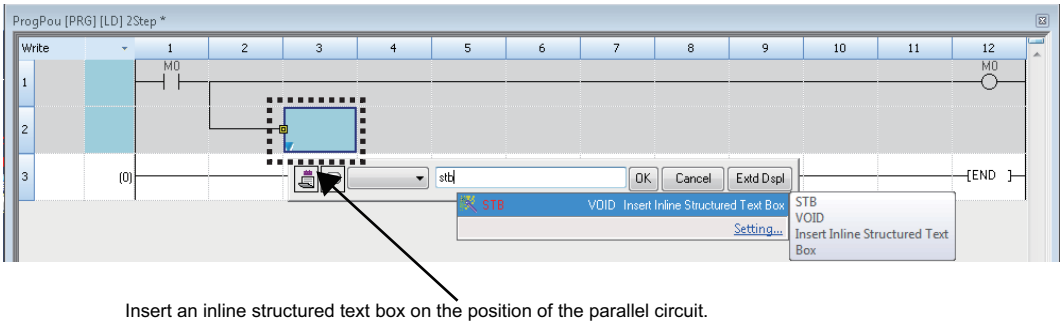
Inline structured text is a function to edit/monitor a program by creating an inline structured text box that displays an ST program, at the coil instruction area on the ladder editor.
With this function, a numeric value operation or a character string process can be created easily in the ladder program.
Up to 2048 characters can be entered in an inline structured text box. (Two characters are used as the line feed.)
Note that this function cannot be used in a safety program and a safety FB/FUN.

Operating procedure

Select [Edit] ⇨ [Inline Structured Text] ⇨ [Insert Inline Structured Text Box] ( ).
The editing method of the inline structured text program is the same as that of Structured Text. (☞ Page 147 Entering programs)

Point

- Enter 'STB' on the element entry dialog to insert an inline structured text box.
- When inserting inline structured text as a parallel circuit, draw lines for the parallel circuit first, and then insert the inline structured text box.



Insert an inline structured text box on the position of the parallel circuit.

Precautions

- Both an FB and inline structured text box cannot be placed in a ladder block.
- Inline structured text box cannot be connected to the input/output of FB instance.
- The label of which data type is pointer cannot be used in the inline structured text.
- If the row that includes an inline structured text box is deleted, the whole ladder block is deleted including an inline structured text.

Pasting from each screen

Paste a label name/device name by dragging and dropping from the editor such as label editor and device comment editor.
By dragging and dropping a bit type label/device in a blank cell, a contact is inserted automatically and its label name/device name is displayed.
Additionally, when dragging and dropping a bit type label/device to the right end of the cell, a coil is inserted automatically and its label name/device is displayed.
A word type device/label can be pasted by dragging and dropping on the operand of an instruction.

Registering undefined labels

When an undefined label is entered, the "Undefined Label Registration" screen is displayed and it can be registered on the label editor.

Checking the duplicated coils

Check the duplicated coils in the same program when entering a coil in the program.

Set whether to check the duplicated coils by setting the following option.

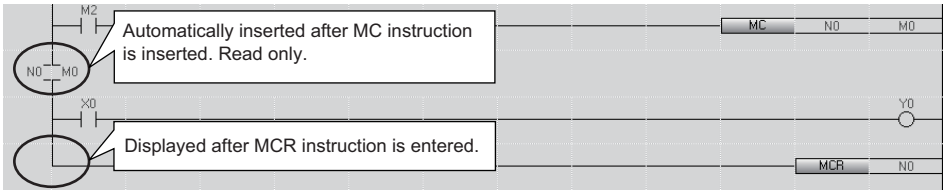
• [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "Ladder Editor" ⇨ "Enter Ladder" ⇨ "Operational Setting"

For the check target instructions/devices of duplicated coils, refer to the following section.

☞ Page 183 Check targets for duplicated coil

Displaying MC/MCR instructions

When the MC instruction is inserted to a ladder program, the double-line symbol is displayed on the left vertical line. As for MCR instruction, the left vertical line is delimited.



Entering comments/statements/notes

This section explains the basic operations for editing comments, statements, and notes.

Entering/editing comments

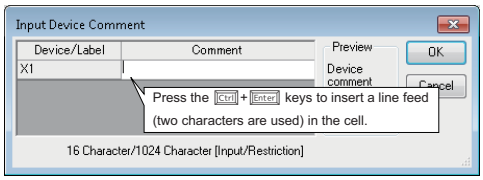
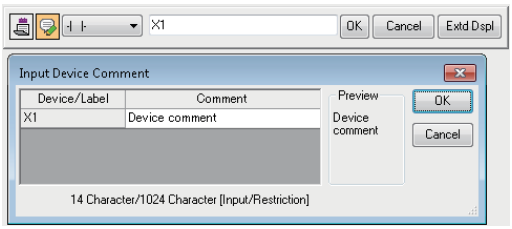
Enter or edit a device/label comment.

To check the entered comments on the ladder editor, change the setting to display comments.

• [View] ⇨ [Comment Display]

The device comments edited or added on the ladder editor are reflected to the device comments set to the following option.

• [Tool] ⇨ [Options] ⇨ "Project" ⇨ "Device Comment Reference/Reflection Target"

Entering/editing method	Operating procedure
On the "Input Device Comment" screen	1. Select [Edit] ⇨ [Documentation] ⇨ [Edit Device/Label Comment] ( ). 2. Select the cell and press the Enter key or double-click it. 3. Enter a comment in the "Comment" column.  After entering comments is completed, select [Edit] ⇨ [Documentation] ⇨ [Edit Device/Label Comment] ( ) again.
On the element entry dialog	1. Select a cell and press the Enter key. 2. Click on the element entry dialog and click the [OK] button. 3. Enter the comment to "Comment" column. 
With the keyboard	1. Display the comments. 2. Select the cell and press the F2 key twice. 3. Enter a comment directly.
From each editor	Label editor: ☞ Page 107 Configuration of label editor Device comment editor: ☞ Page 179 Creating device comments

Entering/editing statements

Add a comment to a ladder block using the statement (line statement, P statement, or I statement) in order to make the processing flow easy to understand.

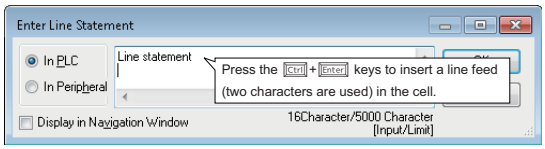
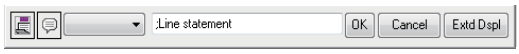
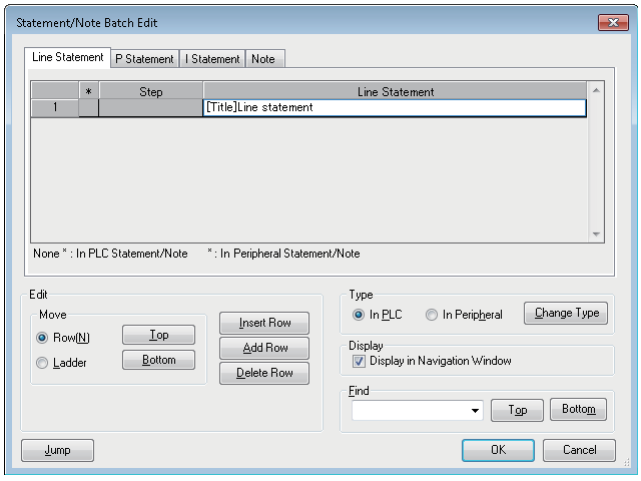
To check the entered statements on the ladder editor, change the setting to display the statements.

• [View] ⇨ [Statement Display]

For details on the statements, refer to the following manuals.

📖 MELSEC iQ-R Programming Manual (Program Design)

📖 MELSEC iQ-F FX5 Programming Manual (Program Design)

Entering/editing method	Operating procedure
On the Enter Statement screen	1. Select [Edit] ⇨ [Documentation] ⇨ [Statement] ( ). 2. Select the cell and press the Enter key, or double-click it. 3. Enter a statement. Before entering P statement/I statement, select the cell of a pointer number or interrupt pointer number.  When "Display in Navigation Window" is selected, "[Title]" is added in front of the line statement. After entering statements is completed, select [Edit] ⇨ [Documentation] ⇨ [Edit Statement] ( ) again.
On the element entry dialog	1. Select the cell and press the Enter key. 2. Enter a statement after entering ';' for 'In PLC' and ';*' for 'In Peripheral'. When displaying a statement on the Navigation window, enter [Title] after ';' or ';*'. For P statement/I statement, enter a statement following the entry of the displayed pointer number or interrupt pointer number. 
On the "Statement/Note Batch Edit" screen	1. Select [Edit] ⇨ [Documentation] ⇨ [Statement/Note Batch Edit] ( ). 2. Enter a statement.  [Insert Row] button: Inserts a row above the cursor position. [Add Row] button: Inserts a row below the cursor position.
With the keyboard	1. Select the cell of a statement, and press the F2 key. 2. Enter a statement directly.

Precautions

';' cannot be prefixed to a line statement.

■Editing statements

After editing a statement, the program will be in the unconverted state.

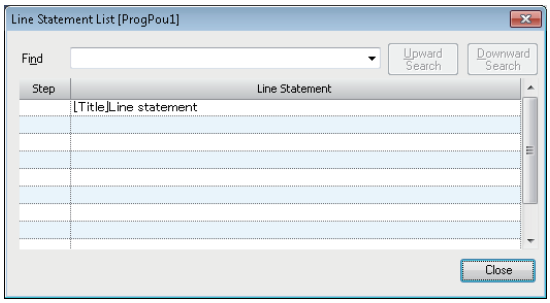
■Displaying list

Display the line statements used in the ladder program in a list.

The cursor is moved to the location where the selected line statement is used.

Window

[Find/Replace] ⇨ [Line Statement List] ( )

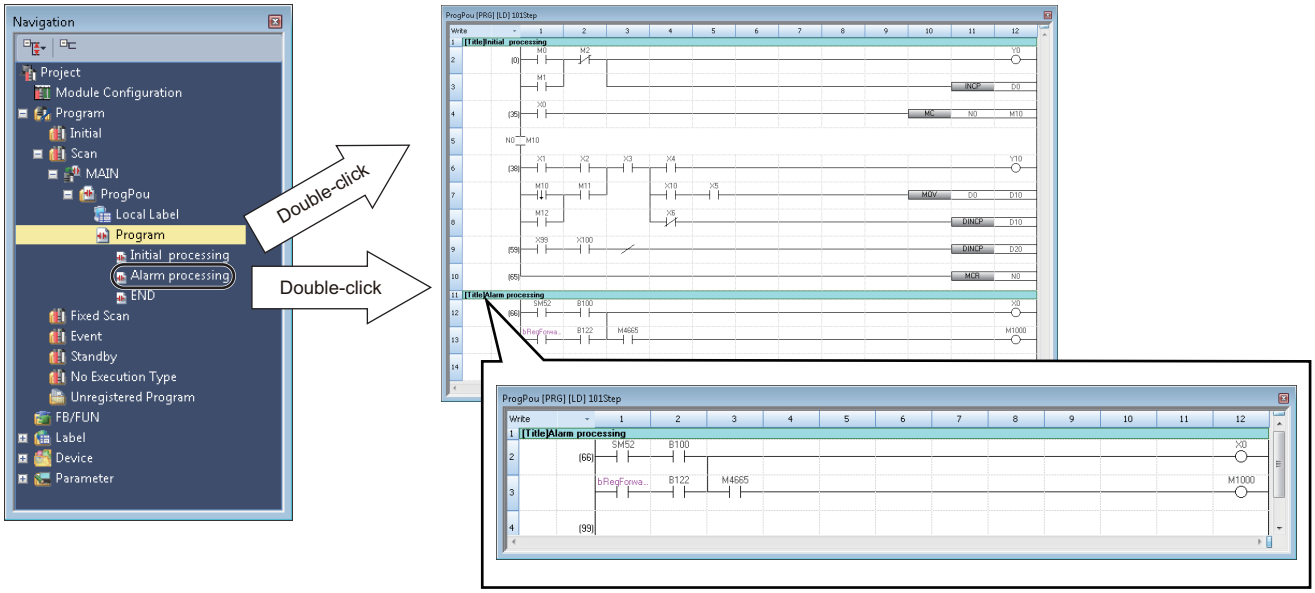


■Line statement to be displayed in the tree

Select a line statement and select [Edit] ⇨ [Document] ⇨ [Show/Hide of Navigation Window] to switch display/hide the line statement on the Navigation window.

When copying/deleting the line statement displayed in the tree, the ladder blocks before the next line statement in the tree are copied/deleted.

Double-click a line statement in the tree to move the cursor to the corresponding ladder block.



■Consideration when using statements in an FB program

When the name of a statement used in an FB program is changed, all programs in which the function block is used in the unconverted state.

Entering/editing notes

Append a comment to coils/application instructions using Note in the program in order to make it easier to understand the content of program.

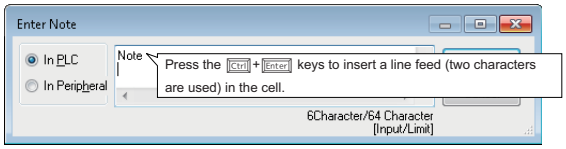
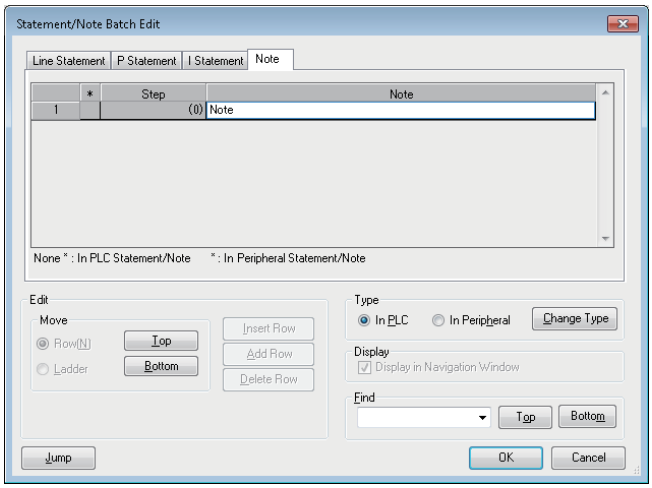
To check the entered notes on the ladder editor, change the setting to display the notes.

• [View] ⇨ [Note Display]

For details on the notes, refer to the following manuals.

📖 MELSEC iQ-R Programming Manual (Program Design)

📖 MELSEC iQ-F FX5 Programming Manual (Program Design)

Entering/editing method	Operating procedure
On the "Enter Note" screen	1. Select [Edit] ⇨ [Documentation] ⇨ [Edit Note] ( ). 2. Select the cell and press the Enter key or double-click it. 3. Enter a note.  After entering notes is completed, select [Edit] ⇨ [Documentation] ⇨ [Edit Note] ( ) again.
On the element entry dialog	1. Press the Enter key. 2. Enter notes following the displayed device/instruction after entering ';' for 'In PLC' and ';*' for 'In Peripheral'.
On the "Statement/Note Batch Edit" screen	1. Select [Edit] ⇨ [Documentation] ⇨ [Statement/Note Batch Edit] ( ). 2. Enter a note.  [Insert Row] button: Inserts a row above the line statement. [Add Row] button: Inserts a row below the line statement.
With the keyboard	1. Select the cell of a note, and press the Enter key. 2. Enter a note directly.

Inserting/deleting NOP instruction

Insert or delete a NOP instruction to adjust Step No. of a program.

The Step No. changed by the NOP instruction is reflected to the ladder editor after converting the program.

Operating procedure

■Batch insertion

1. Select a cell to be insert.
2. Select [Edit] ⇨ [NOP Batch Insert].
3. Set the number of NOPs on the "NOP Batch Insert" screen and click the [OK] button.

■Batch deletion

1. Select [Edit] ⇨ [NOP Batch Delete].
2. Click the [Yes] button.

Copying/pasting elements

Cutting/coping

Select an instruction, range, or ladder block, and cut/copy it.

Select a row when copying a function/function block.

Pasting

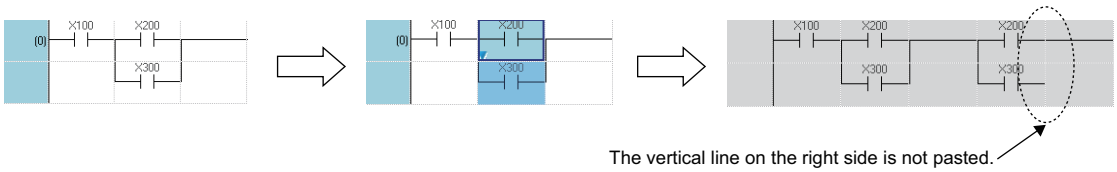
Paste a cut/copied element.

For "Insert mode" (press the Insert key to switch for "Overwrite mode"), the rung is inserted above the position of the cursor.

For "Overwrite mode", the row/column will be inserted automatically and pasted by selecting [Edit] ⇨ [Insert and Paste].

Precautions

If the elements as shown below are selected to set a range, the vertical line on the right side cannot be pasted.



In such case, draw a vertical line after pasting the elements.

Pasting device number/label name consecutively

Increment (+1) and paste a device number and label name consecutively in the cut/copied ladder block.

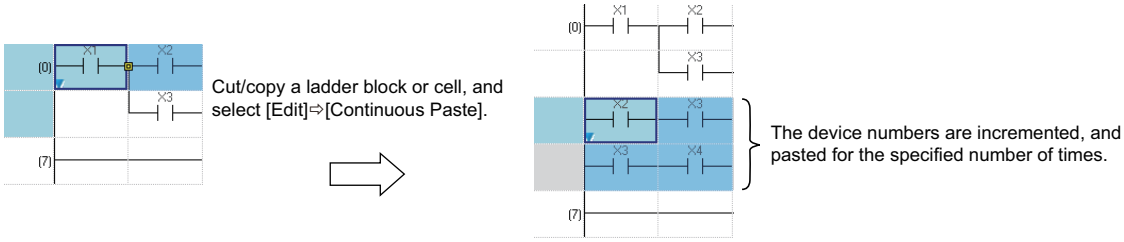
Operating procedure

1. Select [Edit] ⇨ [Continuous Paste].
2. Set the items on the "Continuous Paste" screen, and click the [Execute] button.

Ex.

Set the following on the "Continuous Paste" screen.

Number of Continuous Pastings: 2, Increment Value: 1, Paste direction: Down



Changing ladder blocks temporarily

Change the operation of specific ladder block temporarily.
Using the temporary changing function, the program for debugging can be changed without losing the source program, therefore the debug process will be improved efficiently.

Precautions

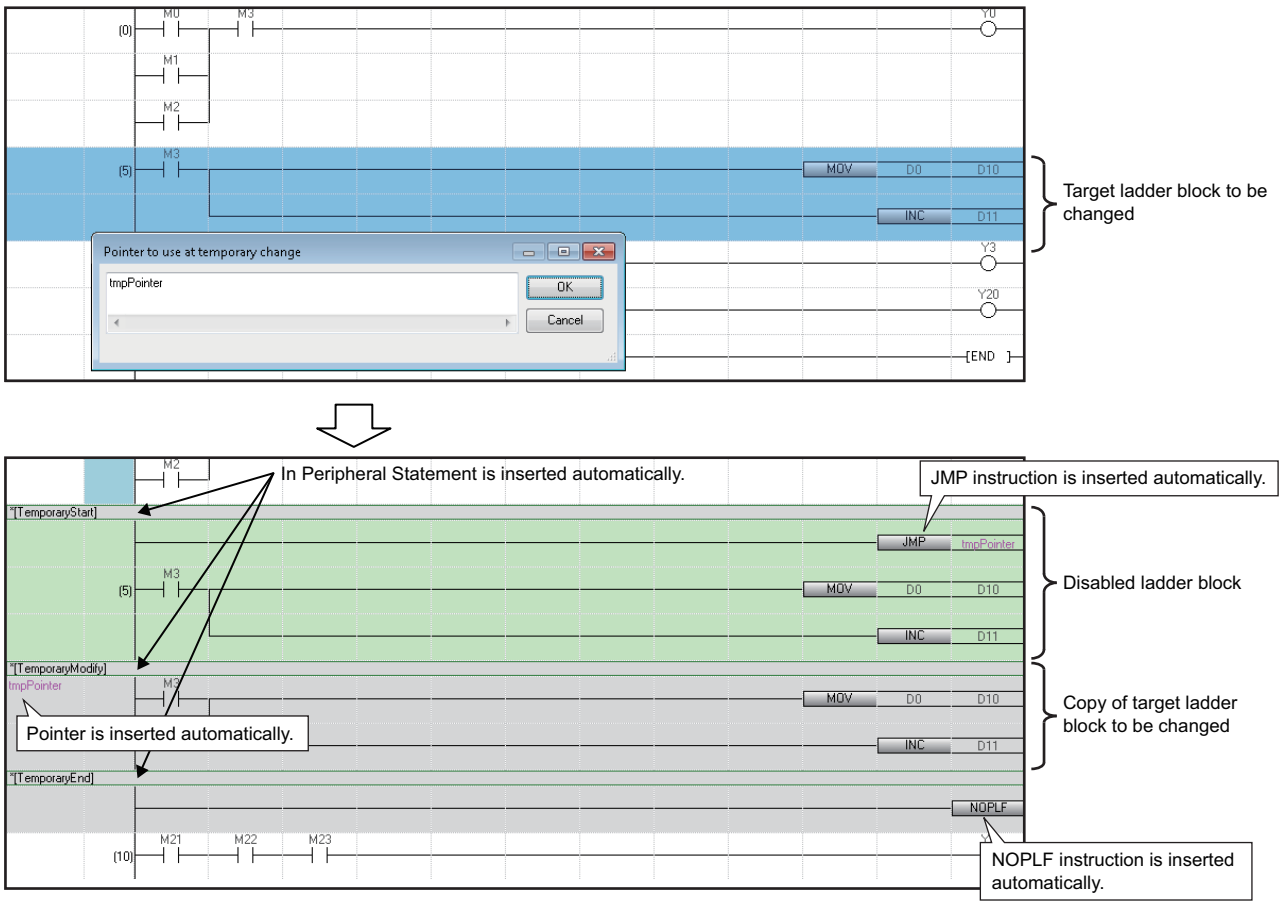
When opening the project, which contains the temporarily changed ladder program, in GX Works2 Version 1.008J or earlier, the background color of disabled ladder block is not changed. Besides, the Apply the Changes operation and the Restore the Changes operation can not be performed.

Temporary changing

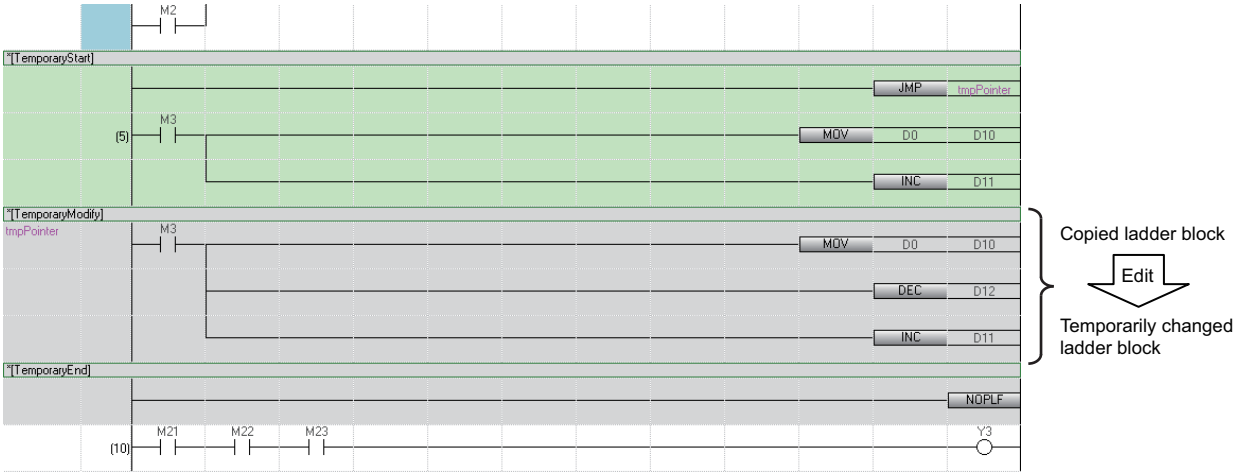
Operating procedure

1. Select the ladder block to be changed, and then select [Edit] ⇨ [Temporarily Change Ladders] ⇨ [Temporarily Change Ladders] ( ).
2. Enter the pointer to be used.

When specifying a label for pointer, create the label in advance.
Peripheral statement/JMP instruction/Pointer/NOPLF instruction are inserted automatically as shown below and the selected ladder block is disabled. Besides, the copy of ladder block is created immediately below the disabled ladder block.

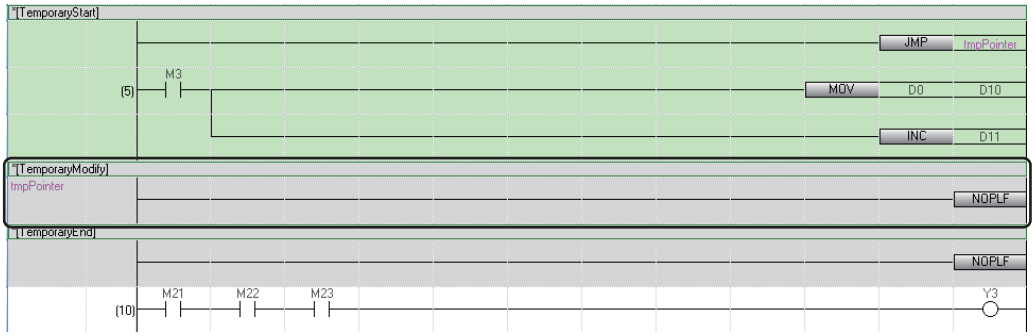

Pointer to use at temporary change
tmpPointer
OK
Cancel
Target ladder block to be changed
In Peripheral Statement is inserted automatically.
JMP instruction is inserted automatically.
Disabled ladder block
Copy of target ladder block to be changed
Pointer is inserted automatically.
NOPLF instruction is inserted automatically.

3. Edit the copied ladder block.



Point

By changing the copied ladder block to the NOPLF instruction, the specified ladder block is disabled temporarily.



Precautions

In a program of function/function block, a pointer device can not be used temporarily. Only pointer type label is applicable.

Applying/restoring the changes

The changed ladder program is applied if no problem is found with the operation. If any problems are found, the change is restored.

Operating procedure

Select the ladder block from [TemporaryStart] statement to [TemporaryEnd] statement, and then select [Edit] ⇨ [Temporarily Change Ladders] ⇨ [Apply the Changes] ( )/[Restore the Changes] ( ).

Considerations for editing the temporarily changed ladder block

When the temporarily changed ladder block was edited with the contents listed in the following table, the program may not be performed normally.

Description	
Statement	Delete/cut a statement.
	Insert a ladder block between the [TemporaryModify] statement row and the temporarily changed ladder block (which is the row setting a pointer).
	Delete the [TemporaryEnd] statement row and NOPLF instruction.
	Insert a ladder block/a statement/NOPLF instruction between the [TemporaryEnd] statement row and NOPLF instruction.
	Insert a statement before and after the [TemporaryStart] statement row.
JMP instruction, Pointer	Edit/delete a JMP instruction or a pointer.
Ladder block	Edit a disabled ladder block.
	Delete a disabled/temporarily changed ladder block.
	Copy and past a disabled/temporarily changed ladder block.

Searching for/replacing data in program

The search functions that can be used in ladder editor are as follows.

Function name	Description	Reference
Simple search for instruction devices/labels	Searches a device/label and moves the cursor on it.	Page 143 Simple search for instruction devices/labels
Jump	Moves the cursor on a specified Step No.	Page 143 Jump
Cross reference	Checks devices, label declaration locations, and referenced locations in a list.	Page 200 SEARCHING DATA
Device list	Checks the usage of devices being used.	
Search and replace	• Finds and replaces a device name, label name, instruction name, and character strings. • Changes open/close contact. • Replaces devices in a batch.	

Simple search for instruction devices/labels

Search for a device/label and move the cursor on the ladder editor.

Operating procedure

1. Press the Space key on the ladder editor.
2. Enter the device/label and click the [Find] button on the "Find" screen.

Jump

Specify a Step No., and move the cursor on a ladder editor.

Operating procedure

1. Select [Find/Replace] ⇨ [Jump].
2. Specify a Step No. on the "Jump" screen, and click the [OK] button.

Point

On the program editor, pressing a numeric key on the keyboard enables to display the "Jump" screen.

Displaying instruction help

Check the instructions used in the ladder program on e-Manual Viewer.

To check the instructions, the files of the corresponding programming manual requires to be registered to e-Manual Viewer.

Operating procedure

1. Select the cell where the target instruction is entered.
2. Press the F1 key.

6.4 Creating ST Programs

This section explains how to create an ST program.
The details on the specification of ST program are described in the following manual. Please read it in advance.
📖 MELSEC iQ-R Programming Manual (Program Design)

Point

The display format and detailed operations for each function can be set by the following option setting.
[Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "ST Editor"

Configuration of ST editor

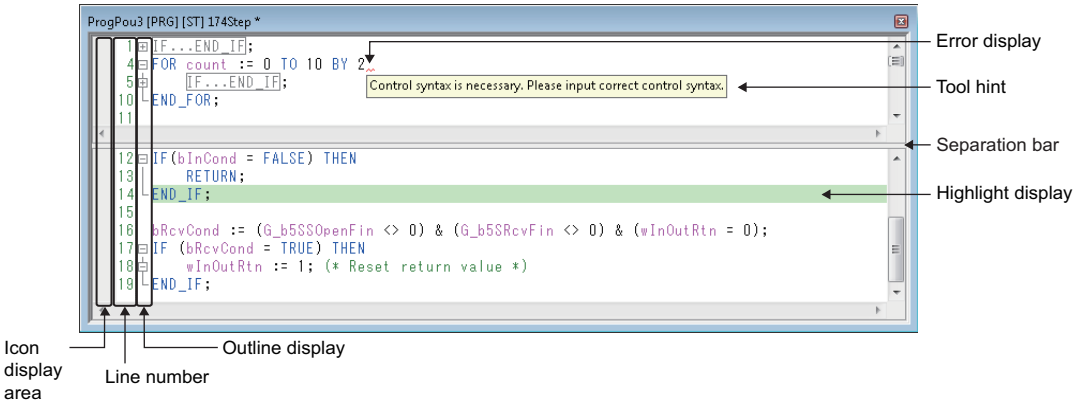
ST editor is a language editor in text format for creating programs in Structured Text.
Spaces, tabs, and line feeds can be entered between keywords and variable names of the ST control syntax.
Terms and symbols that configure the program is referred to as a token.

Window

Project view ⇨ "Program" ⇨ "(execution type)" ⇨ "(program file)" ⇨ "(program block)" ⇨ "Program"

●Toolbar

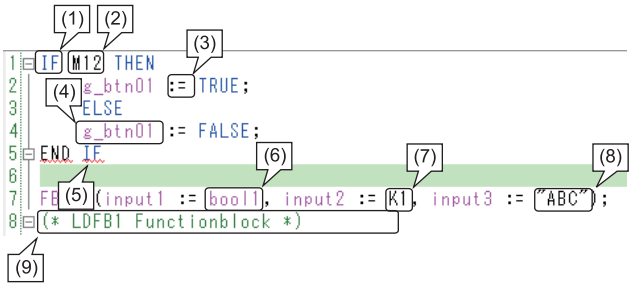
●ST editor



Displayed items

Item	Description	Related operation
Icon display area	The area to display icons.	■Icon type ☞ Page 146 Icon type
Line number	The line numbers of a program.	■Changing display contents [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "ST Editor" ⇨ "Editor Display Items"
Highlighted display	The line on which the cursor is placed is highlighted.	
Error display	A syntax error of the program is displayed.	
Outline display	Symbols to display/hide the text blocks are displayed.	■Show/Hide • [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "ST Editor" ⇨ "Editor Display Items" • [View] ⇨ [Outline] ⇨ [Show/Hide of Outlines]
Tool hint	The information where the mouse cursor is placed over is displayed.	■Changing display contents [Tool] ⇨ [Options] ⇨ [Program Editor] ⇨ "ST Editor" ⇨ "Tool Hint"

■Colors of programs (default)

Figure	Description	Default color
	(1) Syntax	Blue
	(2) Device	Black
	(3) Operator	Black
	(4) Global label	Magenta
	(5) Error	Red
	(6) Local label	Magenta
	(7) Constant	Black
	(8) String constant	Black
	(9) Comment	Green

Point

A font color, background color, and fonts can be changed.

☞ Page 48 Checking and Changing Colors and Fonts

■Icon type

Icon	Description
	This icon is displayed on the line jumped from the Cross Reference window.
	This icon is displayed on the error line jumped from the Output window.

■Keywords for collapse and automatic indent

A line can be outlined and displayed collapsed by using the keywords shown below. Additionally, pressing the Enter key in a control syntax inserts an indent automatically.

Category	Start	End	Automatic indent
Comment texts	(*	*)	×
	/*	*/	×
Selection statement	IF	END_IF	○
	CASE	END_CASE	○
While loop	FOR	END_FOR	○
	WHILE	END_WHILE	○
	REPEAT	END_REPEAT	○

Entering programs

This section explains how to enter an ST program.

For the functions/instructions used in ST programs, refer to the following manuals.

MELSEC iQ-R Programming Manual (Program Design)

MELSEC iQ-F FX5 Programming Manual (Program Design)

Point

Press the Ctrl + Shift + = keys to enter an assignment operator (:=).

Precautions

When an ST editor is set to read-only or monitoring, the program cannot be edited.

Inserting instructions, functions, and control syntaxes

Instructions, functions, and control syntaxes can be entered by entering text using keyboard or by dragging and dropping them.

The input control syntaxes, operators, devices, and TRUE/FALSE are changed to the upper-case characters automatically.

Labels can be entered with aliases.

■Specification method for data type of device

Normally, the data type of a word device is handled as INT (Word [Signed]) on the ST editor.

By adding a suffix that indicates data type (type specifier) to a device name, the devices in which 32-bit integer or real number is stored can be described directly in the operation formula without defining labels.

For details, refer to the following manual.

MELSEC iQ-R Programming Manual (Program Design)

Inserting function blocks

Insert a function block to an ST program.

For the creation methods of a program in a function block, refer to the following section.

☞ Page 214 Creating Function Blocks

Operating procedure

1. Drag and drop a function block to an arbitrary position on an ST editor from the Project view or Element Selection window.
2. Enter the information of a label (FB instance) on the "Undefined Label Registration" screen.
3. Enter the values for the input variables and output variables.

Ex.

When the defined function block is "MYTIMER"

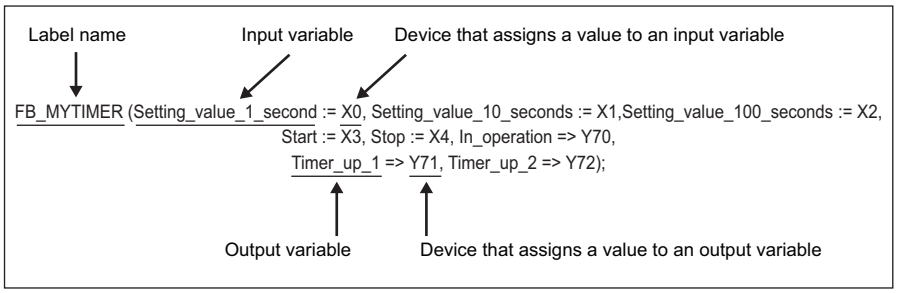
[Details of defined function block]

Label name: FB_MYTIMER

Input variable: Setting_value_1_second, Setting_value_10_seconds, Setting_value_100_seconds, Start, Stop

Output variable: In_operation, Timer_up_1, Timer_up_2

The example of entering ST program are shown below.


Label name
Input variable
Device that assigns a value to an input variable
FB_MYTIMER (Setting_value_1_second := X0, Setting_value_10_seconds := X1,Setting_value_100_seconds := X2,
Start := X3, Stop := X4, In_operation => Y70,
Timer_up_1 => Y71, Timer_up_2 => Y72);
Output variable
Device that assigns a value to an output variable

The output of the function block can be obtained by specifying the output variable name with a period (.) suffixed to the function block name.

Enter an instruction to obtain the output after performing the function block call.

```
Y70: = FB_MYTIMER. In operation;
```

Inserting functions

Insert a function in an ST program.

For the creation methods of FUN programs, refer to the following section.

☞ Page 217 Creating Functions

Operating procedure

1. Drag and drop a function to an arbitrary location on the ST editor from the Project view or Element Selection window.
2. Enter the arguments.

Inserting indention

A tab is inserted as an indentation at the head of a new line automatically when a line feed is inserted during program editing.

The tabulator length can be set in the following option setting.

- [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "ST Editor" ⇨ "Edit Operation"

Inserting comments

Enter a comment which does not affect the program processing. Or, comment out/uncommented the already entered programs in a batch.

Operating procedure

■Entering comments

- When the line is one: Enter a comment after entering slashes '//'.
- When specifying the range: Enclose the comment in '/*' and '*/', or '(*' and '*)'.

■Batch comment out/uncomment of a program

1. Select a range to be commented out or uncommented. (Multiple lines can be selected.)
2. [Edit] ⇨ [Comment Out of Selected Range] ( )/[Disable Comment Out of Selected Range] ( )

Registering undefined labels

Register an undefined label on the label editor.

Operating procedure

1. Enter the label name to be registered on an ST editor.
2. Select the token of a label name, and select [Edit] ⇨ [Register Label] ( ).
3. Set the items on the "Undefined Label Registration" screen, and click the [OK] button.

Displaying syntax templates

Syntax template is a format that defines data type of arguments and format of control syntaxes which are defined in each instruction, function, and operator.
The syntax template of the inserted instruction can be displayed on the ST editor.

Operating procedure

1. Select the token of which syntax is to be displayed.
2. Select [Edit] ⇨ [Display Template] ( ).
3. Enter the arguments in accordance with the displayed template.

Delete the data type name enclosed with '?', and enter a label name or device corresponding to its data type.

Point

Select [Edit] ⇨ [Mark Template (Left)] ( )/[Mark Template (Right)] ( ), or press the Ctrl + Alt + ←/→ keys to select an argument of the template one by one.

Searching for/replacing data in program

The search functions that can be used in ST editor are as follows.

Function name	Description	Reference
Jump	Specifies a line number on an ST editor, and move to the corresponding line.	Page 150 Jump
Cross reference	Checks devices*1, label declaration locations and referenced locations in a list.	Page 200 SEARCHING DATA
Device list	Checks the usage of devices being used*1.	
Search and replace	• Searches for and replaces a device name*1, label name, instruction name, and character strings. • Changes open/close contact. • Replaces devices in a batch.	

*1 Search with a device name except for the device suffix part.

Jump

Specify a line number and move the cursor on a ST editor.

Operating procedure

1. Select [Find/Replace] ⇨ [Jump].
2. Enter a line number of the program on the "Jump" screen, and click the [OK] button.

Displaying instruction help

Check the instructions used in an ST program on e-Manual Viewer.

To check the instructions, the files of the corresponding programming manual requires to be registered to e-Manual Viewer.

The applicability of help display is shown below.

Token type	Applicability
Operator	×
Control syntax	×
FB	×
Function	○
Constant, variable, comment	×

Operating procedure

1. Place the cursor on the token of the target instruction.
2. Press the F1 key.

6.5 Creating FBD/LD Programs

This section explains how to create an FBD/LD program.
The details on the specification of FBD/LD program are described in the following manual. Please read it in advance.
📖 MELSEC iQ-R Programming Manual (Program Design)

Point

The display format and detailed operations for each function can be set by the following option setting.
[Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "FBD/LD Editor"

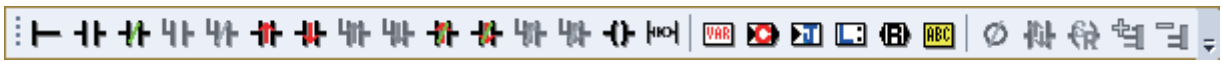
Configuration of FBD/LD editor

FBD/LD editor is a graphical language editor to create programs by combining Function Block Diagram language with Ladder Diagram language.
Programs can be created only by placing the prepared elements vertically and horizontally, and connecting them.

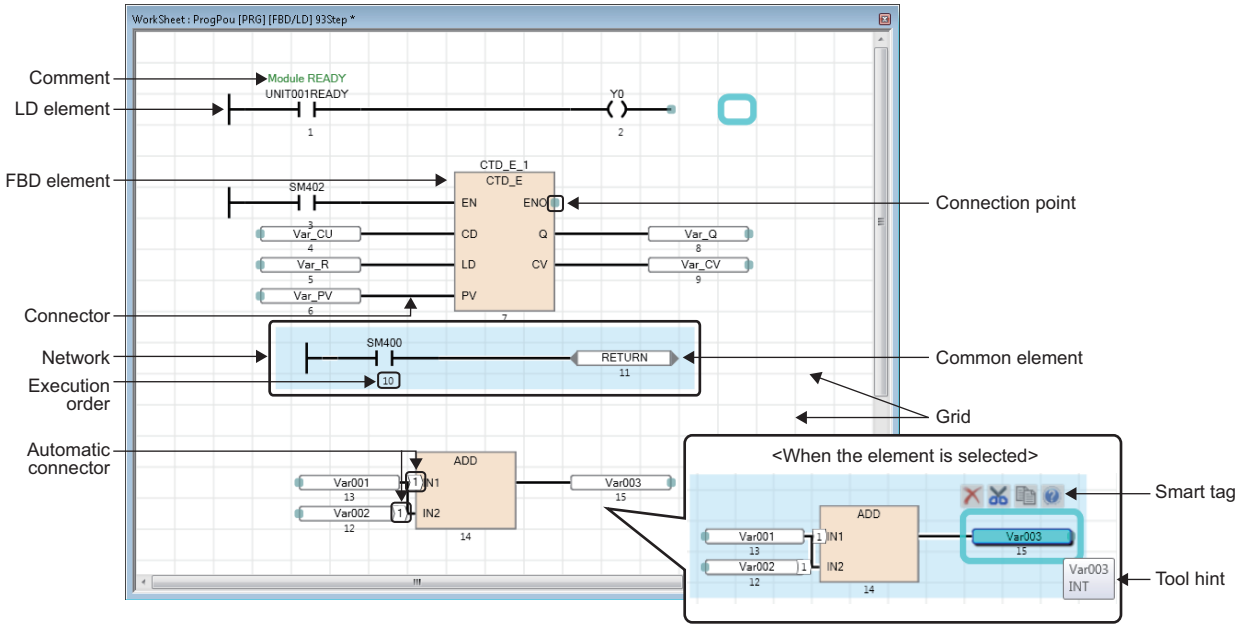
Window

Project view ⇨ "Program" ⇨ "(execution type)" ⇨ "(program file)" ⇨ "(program block)" ⇨ "(worksheet)"

●Toolbar



●FBD/LD editor



Displayed items

Item	Description	Related operation
Comment	A comment set to a device/label.	■Show/Hide [View] ⇨ [Comment Display]
LD element	An element that constitutes a ladder program.	■Element type ☞ Page 152 LD element
FBD element	An element that constitutes an FBD program.	■Element type ☞ Page 153 FBD element
Common element	A common element used regardless of program language.	■Element type ☞ Page 153 Common element
Connector	A line that connects the connection points between elements. It is displayed by placing an element to be connected closer to the connection point on the connection target.	—

Item	Description	Related operation
Network	A single network (a single block) composed of all elements connected together.	■Releasing from a network Select the element to be released, and move it with the Shift key held down.
Execution order	The execution order of a program is displayed.	■Show/Hide [View] ⇨ [Display Execution Order]
Automatic connector	When a connector cannot be displayed due to the conditions where an element is placed, a number is displayed automatically. The same number of automatic connector indicates that they are connected.	—
Connection point	A terminal point to connect elements with a connector. By adding an element while a connection point is being selected, the element can be added with it connected already.	■Inverting contact ☞ Page 154 Switching methods for contacts/instructions
Grid	A grid line that serves as makers to place the elements.	■Show/Hide [View] ⇨ [Grid Display]
Smart tag	An operation button displayed around the selected element. Relevant functions can be performed by clicking the button.	■Show/Hide [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ FBD/LD Editor" ⇨ "Smart Tag"
Tool hint	The information on device/label and FB/FUN where the mouse cursor is placed over is displayed.	■Changing display contents [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "FBD/LD Editor" ⇨ "Tool Hint"

Point

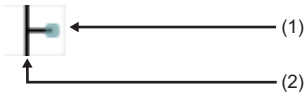
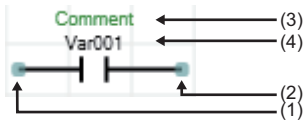
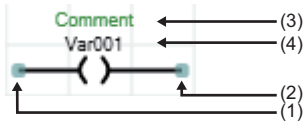
A font color, and fonts can be changed. However, this dose not apply to the comment elements.

☞ Page 48 Checking and Changing Colors and Fonts

Elements

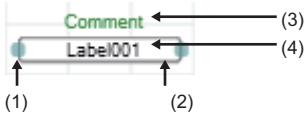
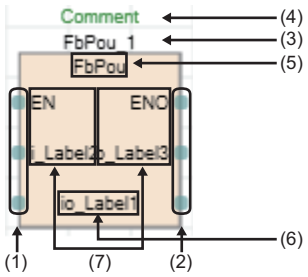
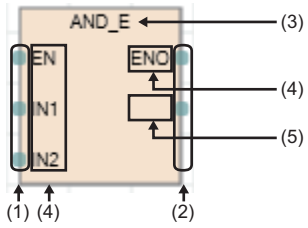
The elements that can be used in FBD/LD program are shown below:

■LD element

Element	Name of each part	Description
Left power rail element 	(1) Output connection point (2) Left power rail	The position of a left power rail can be placed at will, and it will be the starting point for creating a ladder program.
Contact element 	(1) Input connection point (2) Output connection point (3) Label comment/device comment*1 (4) Device/label	Specify the device/label. It transmits the ON/OFF signal according to the specified information. For details on the element, refer to the following manuals. MELSEC iQ-R Programming Manual (Program Design) MELSEC iQ-F FX5 Programming Manual (Program Design)
Coil element 		Specify the device/label. It outputs the signal to the specified device/label according to the transmitted ON/OFF signal. For details on the element, refer to the following manuals. MELSEC iQ-R Programming Manual (Program Design) MELSEC iQ-F FX5 Programming Manual (Program Design)

*1 Displaying/hiding a comment is switched by the following option setting.
[Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ FBD/LD Editor" ⇨ "Comment" ⇨ "Display Label/Device Comment"

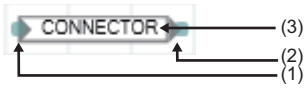
■FBD element

Element	Name of each part	Description
Variable element 	(1) Input connection point (2) Output connection point (3) Label comment/device comment*1 (4) Device/label	Specify the device/label. Information can be obtained/stored in the specified device/label. It is switched to a constant element by entering a constant.
Constant element 	(1) Output connection point (2) Constant value	Specify a constant. The specified constant can be output. It is switched to a variable element by entering a device/label.
Function block element 	(1) Input connection point (2) Output connection point (3) FB instance name (label) (4) Label comment (5) Data type (6) Input/output label (VAR_IN_OUT) (7) Input/output label (other than VAR_IN_OUT)	Indicates a block for function corresponding to the data type. It is used with an FB instance name assigned to each element. For details on the element, refer to the following section. ☞ Page 213 SEGMENTING PROGRAMS
Function element 	(1) Input connection point (2) Output connection point (3) Data type (4) Input/output label (argument) (5) Return value	Indicates a block for function corresponding to the data type. The name is not displayed in the return value. For details, refer to the following section. ☞ Page 213 SEGMENTING PROGRAMS

*1 Displaying/hiding a comment is switched by the following option setting.
[Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ FBD/LD Editor" ⇨ "Comment" ⇨ "Display Label/Device Comment"

■Common element

For a jump element and a return element, inverting contact on its connection point is not available.

Element	Name of each part	Description
Jump element 	(1) Input connection point (2) Label*1	It is used to jump the execution processing from a jump element to a jump label element.
Jump label element 	(1) Label*1	Enter a label to be specified as the jump destination.
Connector element 	(1) Input connection point (2) Output connection point (3) Connector label	It is used instead of a connector when placing a network in the display range/print range on the editor. The same connector label indicates that they are connected.
Return element 	(1) Input connection point (2) Character string, "RETURN" (Uneditable)	It is used to suspend the processing in the middle of processing.
Comment element 	(1) Comment display area	It is used to enter a comment. Double-click it with a mouse cursor placed on the frame of the element in order for the size to be adjusted automatically according to the character string.

*1 Only local label of which data type is pointer can be specified.

Entering programs

This section explains how to enter an FBD/LD program.

Adding elements

Operating procedure

■Using the edit box

Select[*1] a cell that the element is to be added, then enter a label name or data type of FB/FUN directly.

An element name and label name can be edited directly by selecting a placed element and pressing the F2 key.

The following items can be entered:

- Device/label
- Constant
- FB/FUN

*1 The options of instruction/label can be displayed by pressing the Ctrl + Space keys.

■Inserting from the menu/toolbar

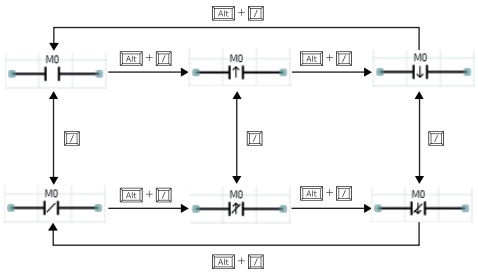
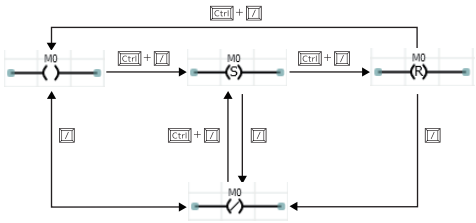
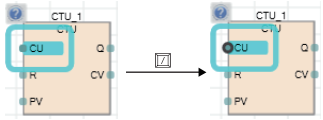
Select a cell to which an element is to be added on the FBD/LD editor, then select [Edit] ⇨ [Add Element] ⇨ [(element)]. It can also be added by selecting from the toolbar.

■Inserting from the Element Selection window

Select an element from the Element Selection window, and drag and drop it on the FBD/LD editor.

■Switching methods for contacts/instructions

Select a contact/instruction to be switched, and select [Edit] ⇨ [Easy Edit] ⇨ [Invert Contact (Open/Close)]( )/[Switch Pulse]( )/[Switch SET and RST]( ) or follow the methods shown below.

Switching contacts and instructions	Shortcut key	Remarks
Switching open/close contact, raising/falling pulse		—
Inverting contact, switching SET/RST instructions		—
Inverting contact on connection point of FB/FUN		Only when the data type of the connection point is one of the following: • Bit • Word [Unsigned]/Bit String [16-bit] • Double Word [Unsigned]/Bit String [32-bit] • ANY_BIT • ANY_BOOL

■Specification method for data type of device

On the FBD/LD editor, word device as the data type can be specified.

By adding a suffix that indicates data type (type specifier) to a device name, the data type can be identified.

For details, refer to the following manual.

📖 MELSEC iQ-R Programming Manual (Program Design)

Common operations of elements

Operation		Operating procedure	
		Mouse	Keyboard
Select	To select a single element	Click an element to be selected.	Move the cursor on an element to be selected.
	To select multiple elements	• Click multiple elements with the Ctrl/Shift key held down. • Click the background of the FBD/LD editor, and drag it diagonally.	Select multiple elements by moving the cursor with the Shift key held down.
	To select a network	Click an element, and select [Edit] ⇨ [Select Network].	Select an element and press the Ctrl+Shift+A keys.
	To select whole elements	The operation is same as the one for selecting multiple elements.	Press the Ctrl+A keys.
Change a name		Double-click the element of which name is to be changed, and enter a name.	Select the element of which name is to be changed, and press the Enter*1/F2 key, or enter a name.
Move		Click the element to be moved, and drag and drop it on the desired position. (By moving an element with the Shift key held down, it can be separated from the network and moved.)	Select the element to be moved, and press the Ctrl+Shift+←/→/↑/↓ keys.
Connector	To connect	• Click a connection point, and drag it to a target connection point. • Click the element to be connected, and drag it closer to a target connection point.	—
	To replace	—	Select a connector with the Shift key held down, and drag and drop it on a connection point of an element.
	To insert an element	Click an element, and move it to a connected connector with the Shift key held down. (Only an element which has a connection point on the same level at the input/output side)	—

*1 For function element and function block element, this follows the operation for double-clicking, which is selected from [Tool] ⇨ [option] ⇨ "Program Editor" ⇨ "FBD/LD Editor".

Replacing function elements/function block elements

Operating procedure

1. Select a function element/function block element.
2. Select [Edit] ⇨ [Change FB/FUN Data], and enter the data type to be changed.

Point

The element also can be replaced by dragging an element from the Element Selection window and dropping it on the element to be changed.
For function elements, similarly, the data type can be changed by selecting a function element and entering a new data type directly.

Adding/deleting arguments

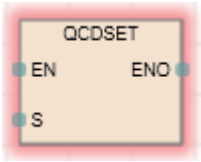
Only for functions of which number of arguments can be changed, the arguments can be added/deleted.

Operating procedure

1. Move the cursor on a function element.
2. Select [Edit] ⇨ [I/O Argument] ⇨ [Increment Argument]( )/[Delete Argument]( ).

Updating undefined FBs/FUNs

When a definition of a pasted element is unsure due to its deletion or change, or an element without the definition information is pasted, it is displayed as shown below resulting in an error.
Select [Edit] ⇨ [Update FB/FUN], and update the definition information.
When the data type of FB/FUN has been changed, select [Edit] ⇨ [Update FB/FUN].



Adjusting position automatically

The position is automatically adjusted to avoid overlap of elements when an element is added/moved.
When a rectangle area of network (an area comprises elements connected with connectors and additional one cell to each direction) overlaps with other area, the position is automatically adjusted so that the entire network does not overlap.
However, comment elements can be superimposed since it is not subject for the automatic adjustment.

Inserting/deleting rows and columns

Operating procedure

■Inserting a row

Select a cell in the row to be inserted, then select [Edit] ⇨ [Insert Row].
A row will be inserted on the selected cell.

■Deleting a row

Select a cell in the row to be deleted, then select [Edit] ⇨ [Delete Row].
The row that contains the selected cell will be deleted. If the elements exist in the row that is same as the selected cell, the row will not be deleted.

■Inserting/deleting a column

Move the cursor in the network that a column will be inserted/deleted, then select [Edit] ⇨ [Insert Column of Network]/[Delete Column of Network].
A column is inserted/deleted within the range of network.

Pasting from each screen

Paste a label name/device name by dragging and dropping from the label editor and the device comment editor.

Registering undefined labels

When an undefined label is entered, the "Undefined Label Registration" screen is displayed and it can be registered on the label editor.

Searching for/replacing data in programs

The search functions that can be used in FBD/LD editor are as follows.

Function name	Description	Reference
Cross reference	Checks devices, label declaration locations, and referenced locations in a list.	Page 200 SEARCHING DATA
Device list	Checks the usage of devices being used.	
Search and replace	• Searches for and replaces a device name, label name, instruction name, and character strings. • Changes open/close contact. • Replaces devices in a batch.	

Displaying help

Check the elements used in the FBD/LD program on e-Manual Viewer.

To check, the files of the corresponding programming manuals requires to be registered in e-Manual Viewer.

Operating procedure

1. Select the element to be checked.
2. Press the F1 key.

6.6 Creating SFC Programs

This section explains how to create an SFC program.
The details on the specification of SFC program are described in the following manual. Please read it in advance.
📖 MELSEC iQ-R Programming Manual (Program Design)

Point

The display format and detailed operations for each function can be set by the following option setting.
[Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "SFC Diagram Editor"

Creation procedure

1. Set the point of step relay(S) in the detailed setting in the "Device/Label Memory Area Detailed Setting" of CPU parameter. (The default point of step relay(S) is 0 points.)
2. Create new SFC data. (☞ Page 72 Creating projects)
3. Configure the following settings if needed.

- the "SFC Setting" of CPU parameter.
- The "Act at Block Multi-Activated" setting to program files
- The "SFC information device"to blocks (Block information)

For details, refer to the following manual.
📖 MELSEC iQ-R Programming Manual (Program Design)

4. Open the SFC diagram editor, then create an SFC diagram. (☞ Page 163 Creating SFC diagrams)
5. Edit a program of action/transition. (☞ Page 172 Creating/displaying Zooms (action/transition))

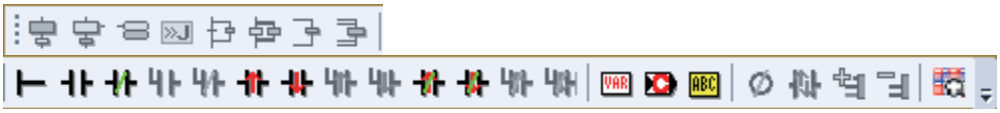
Configuration of SFC diagram editor

The SFC diagram editor is a graphical language editor to show a sequence control as a state transition diagram.
By simply inserting the prepared SFC elements along the operation flow, the each element is connected automatically and a program can be created.

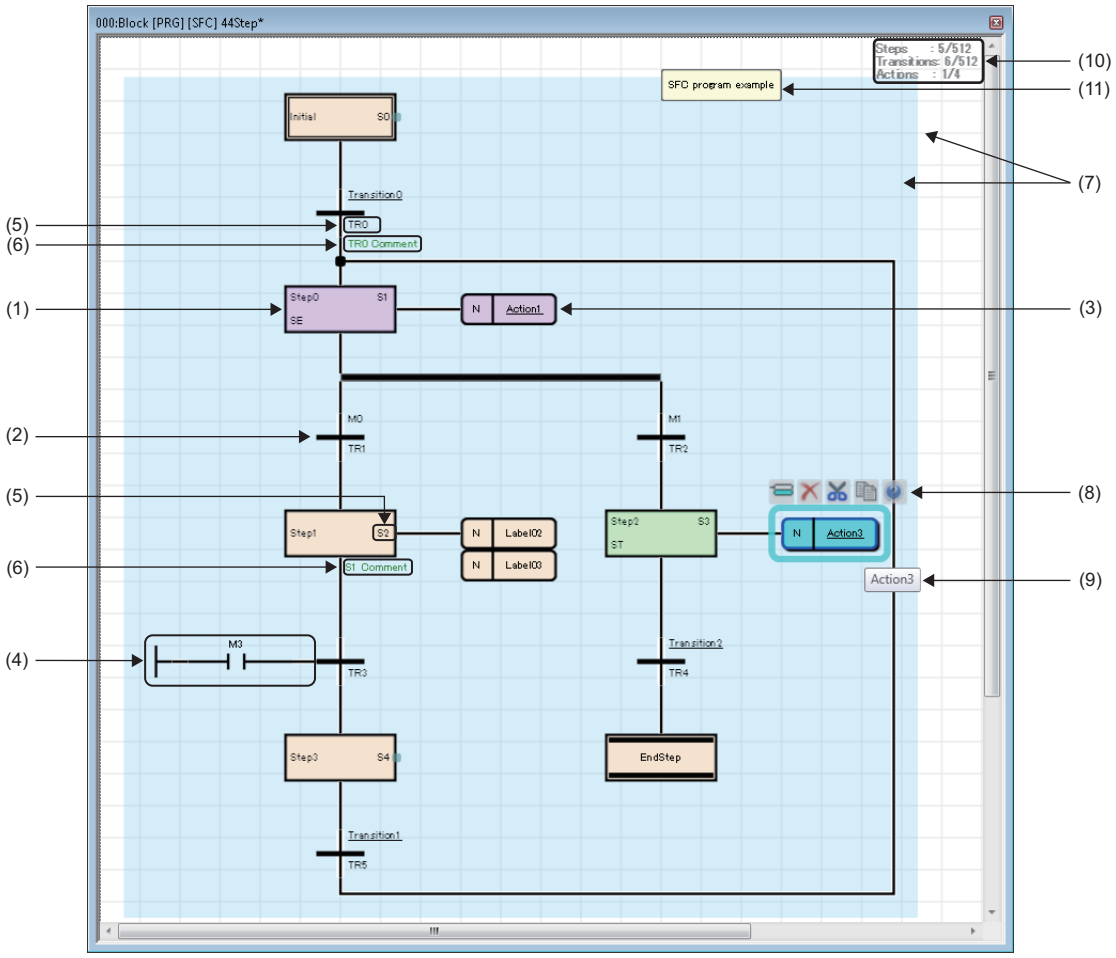
Window

Select "Program" ⇨ "(execution type)" ⇨ "(program file)" ⇨ "(block)" ⇨ "Program" on the Project view.
Immediately after creating a program, SFC elements that are necessary to create a program such as an initial step and end step are placed.

●Toolbar



●SFC diagram editor



Displayed items

Item		Description	Related operation
SFC element	(1) Step	Indicates one processing of a program.	—
	(2) Transition	Indicates a condition (transition conditions) to move to the next step. Transitions are written in a Zoom or on an SFC diagram, and they can be displayed in multiple formats. For details, refer to the following section. ☞ Page 162 Display format of action/transition Ladder Diagram, Structured Text, and Function Block Diagram / Ladder Diagram can be used in a Zoom.	■Displaying a start destination block [View] ⇨ [Open Zoom/Start Destination Block]
	(3) Action	Indicates an assigned operation output to a step. Actions are written in a Zoom or an SFC diagram, and they can be displayed in multiple formats. For details, refer to the following section. ☞ Page 162 Display format of action/transition Ladder Diagram, Structured Text, and Function Block Diagram / Ladder Diagram can be used in a Zoom.	■Displaying a start destination block [View] ⇨ [Open Zoom/Start Destination Block]
(4) FBD/LD element		An FBD/LD element that can be used only for a transition. The elements that can be used on an SFC diagram differs from the one for an FBD/LD editor. For details, refer to the following manual. 📖 MELSEC iQ-R Programming Manual (Program Design)	■FBD/LD element type ☞ Page 152 Elements ■Editing method ☞ Page 154 Entering programs
(5) Step No./Transition No.		An assigned number to a step/transition automatically by conversion. 'S' (Step relay) of CPU module are assigned to steps. Step relays (S☐) are used for SFC control instructions, the current value change in monitor/Watch window, and the data logging/Memory Dump function. The assigned number can be changed.	■Show/Hide [View] ⇨ [Display Step/Transition] ■Changing assigned numbers ☞ Page 172 Editing Step No./Transition No.
(6) Device comment		The device comment of a Step No. (S☐)/Transition No. (TR☐) is displayed. The device comment which was set to transitions/actions is not displayed.	■Input ☞ Page 163 Entering comments of Step No., Page 164 Entering comments of /Transition No. ■Show/Hide [View] ⇨ [Comment Display]
(7) Grid		A grid line that serves as makers to place the elements.	■Show/Hide [View] ⇨ [Grid Display]
(8) Smart tag		An operation button displayed around a selected element. Relevant functions can be performed by clicking the button.	■Show/Hide [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "SFC Diagram Editor" ⇨ "Smart Tag"
(9) Tool hint		Information where the mouse cursor is placed over is displayed.	■Changing display contents [Tool] ⇨ [Options] ⇨ "Program Editor" ⇨ "SFC Diagram Editor" ⇨ "Tool Hint"
(10) Information area		"Number of used/maximum number" of SFC elements are displayed. The font color is changed in red when the number of used reached to the maximum number. SFC elements cannot be created once the number reached to the maximum number. In that case, arrange the number of elements, for example, by deleting.	—
(11) Comment		Comments can be put on an SFC editor. The comments does not affect to the program. The frame size of a comment can be adjusted to match to the comment length by placing the mouse cursor on the frame and double-clicking it.	—

Point

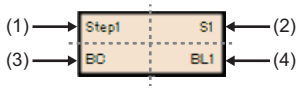
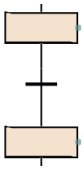
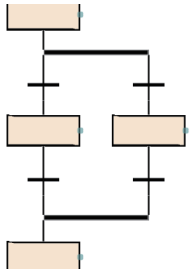
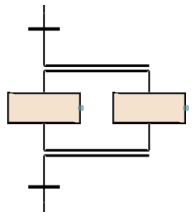
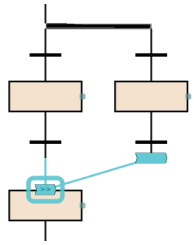
A font color, background color, and fonts can be changed.
☞ Page 48 Checking and Changing Colors and Fonts

SFC element

The following shows elements which can be used in an SFC program.

For details on creatable numbers of block/each element, action, and each element, refer to the following manual.

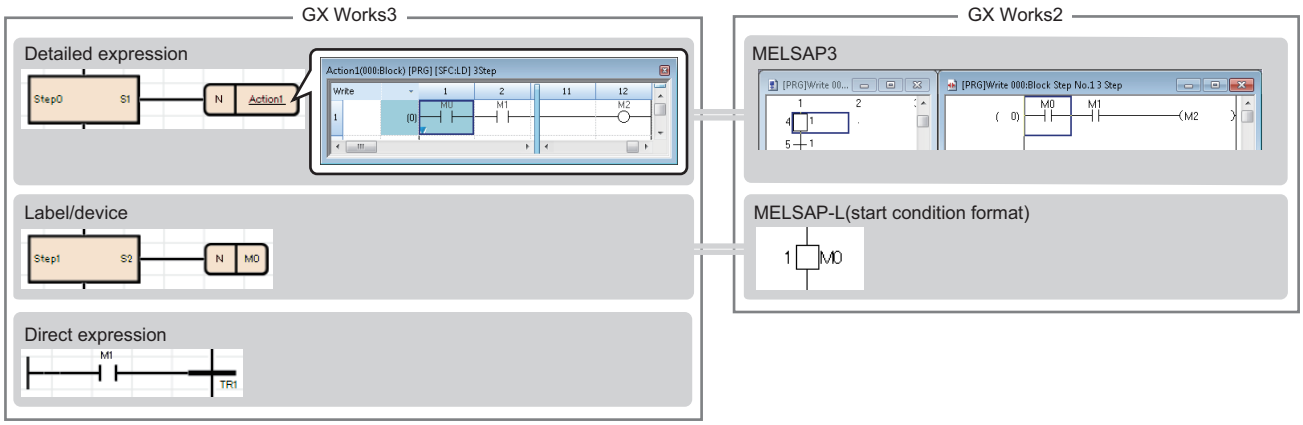
MELSEC iQ-R Programming Manual (Program Design)

Items		Description
Step  (1) Step name (2) Step No. (S□) (3) Step attribute (4) Step attribute target	■Initial step 	Indicates the head of a block. An initial step is necessary in a block. The operation of a step can be changed by specifying the step attribute. For details, refer to the following manual. MELSEC iQ-R Programming Manual (Program Design)
	■Normal step 	Once a transition that places under a step is satisfied, the next step is activated. The operation of a step can be changed by specifying the step attribute. For details, refer to the following manual. MELSEC iQ-R Programming Manual (Program Design)
	■End Step 	Indicates the end of a block. Step No. is not assigned to an end step.
Transition (1) Transition name (2) Transition No. (TR□)	■Series transition 	Moves the processing to a next step.
	■Selection branch 	Indicates a branch displayed with a single line for selecting one processing.
	■Simultaneous branch 	Indicates a branch displayed with a double line for performing multiple processing simultaneously.
	■Jump 	Makes the execution processing jump to the specified step in the same SFC block. A jump and a connection line can be switched.
Action (1) N: Qualifier (2) Action name		Once the steps are activated, the assigned action will be executed. 'N' indicates that an action perform when the step is activated. Other than 'N' cannot be set.

Display format of action/transition

There are multiple display formats (types) for actions/transitions on the SFC editor in GX Works3.
Programs can be shown in the equivalent display formats to MELSAP3 and MELSAP-L (start condition format) in GX Works2 by changing the display formats (type).

Type	Target SFC element	Display format in GX Works2	Reference
Detailed expression	Action, transition	MELSAP3	Page 164 Creating a transition Page 165 Creating an action
Label/device		MELSAP-L (start condition format)	
Direct expression	Transition only	—	



- Action name/transition name is underlined when selecting the detailed expression.
- "*" is displayed to the transition name that is not connected with an FBD/LD element

Creating SFC diagrams

This section explains how to create an SFC diagram.
Elements which can be inserted are different depending on selected places.
Since the size or place of each element/connection line are placed automatically, it cannot be freely changed.

Precautions

When the contents shown below were entered/selected, a red frame appears in the edit box and the settings cannot be configured.

- Duplicate step name/Step No./Transition No.
- Unusable step attribute
- Step name which cannot be specified as a jump destination

Inserting/editing steps

■Inserting a step

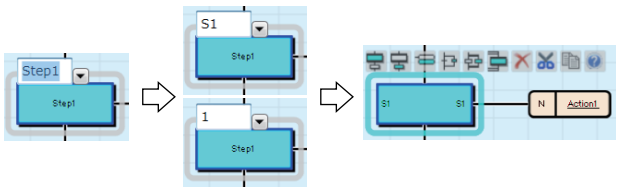
Select a step/transition/jump, then select [Edit] ⇨ [Insert] ⇨ [Step] ( ).

■Changing a step name/step.No./step attribute/step attribute target

- Using a shortcut key (Step names only can be changed):
  Select a step, and press the F2 key.
- On the "Step Properties" screen:
  Select a step, then select [Edit] ⇨ [Properties].
- From the menu:
  Select a step, then select [Edit] ⇨ [Modify] ⇨ [(each item)].
- Double-click:
  Double-click a step. The target to be changed is different depending on the place double-clicked (☞ Page 161 SFC element).

Point

By entering a Step No./number to the edit box for changing a step name, the entered number is set as the step name or the Step No.



■Changing a normal step to an end step

Select a step to be chenged, then select [Edit] ⇨ [Modify] ⇨ [End Step/Jump].
All SFC diagram written below the changed step are deleted.

■Entering comments of Step No.

Enter a comment on the "Step Properties" screen/the device comment editor.

■Creating a step that makes another block activated

A step that activates another block when a step was activated (block start step) is creatable.

1. Insert a step.
2. Specify 'BC' or 'BS' for the step attribute.
3. Specify a block No. to be activated for the step attribute target.

To check the step, which is to be the specification source, from the block specified as the step attribute target, select [View] ⇨ [Open Start Source Block].
For details on the performance of step attribute (BC/BS), refer to the following manual.
📖 MELSEC iQ-R Programming Manual (Program Design)

■Creating a step that makes a step being held deactivated

A step that deactivates a step being held when a step was activated (reset step) is creatable.

1. Insert a step.
2. Specify 'R' for the step attribute.
3. Specify a step name to be deactivated for the step attribute target.
   When 'S999' is specified to a step name, all steps being held will be the target.

For details on the performance of step attribute (R), refer to the following manual.

📖 MELSEC iQ-R Programming Manual (Program Design)

Inserting/editing transitions

■Inserting a transition

Select a step/transition/jump, then select [Edit] ⇨ [Insert] ⇨ [Transition] (⊣).

■Changing a transition name/Transition No.

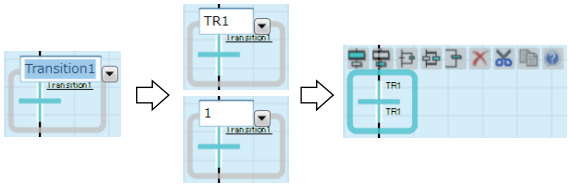
- Using a shortcut key (Transition names only can be changed):
  Select a transition, and press the F2 key.
- On the "Transition Properties" screen:
  Select a transition, then select [Edit] ⇨ [Properties].
- From the menu:
  Select a transition, then select [Edit] ⇨ [Modify] ⇨ [Name]/[Device].
- Double-click:
  By double-clicking a Transition No. (☞ Page 161 SFC element), it can be changed. By double-clicking a transition name, the Zoom is created/displayed.

For a transition that a Zoom has been created, the data name of Zoom will not be changed if the transition name was changed.

Modify the data name to match with the transition name on the "Zoom List" screen. (☞ Page 172 Displaying a Zoom list)

Point

By entering a Transition No./number to the edit box of which the transition name is to be changed, the entered values are set as the transition name and the transition No.



■Creating a transition

There are four ways of creation method for transitions.

Each transition has a different type. The type can be checked on the "Transition Properties" screen.

Creation method		Operating procedure	Type
In a Zoom	Create conditions with a program	☞ Page 172 Creating/displaying Zooms (action/transition)	Detailed expression
On an SFC diagram	Use TRUE/FALSE as conditions	Select a transition, then enter TRUE/FALSE in the transition name.	Label/device
	Use ON/OFF of bit device/bit type label as conditions	Select a transition, then enter a bit device/bit-specified word device, or a bit type label in the transition name.	Label/device
	Create a condition with FBD/LD element	Select [Edit] ⇨ [Modify] ⇨ [Direct Expression for Transition], then connect with an FBD/LD element.	Direct expression

■Entering comments of /Transition No.

Enter a comment on the "Step Properties" screen/device comment editor.

Inserting/editing actions

■Inserting an action

Select a step/an action, then select [Edit] ⇨ [Insert] ⇨ [Action] ( ).

Multiple actions can be inserted in a step.

Point

If all actions in a step are deleted, a transition is only checked all the time while a step is activated. Once the transition was satisfied, the next step is activated.

Steps without actions can be used when synchronizing the steps that are performing simultaneously etc.

■Changing an action name

Click the action, then select [Edit] ⇨ [Modify] ⇨ [Name]/press the F2 key.

For an action that a Zoom has been created, the data name of Zoom will not be changed if the action name was changed. Modify the data name to match with the action name on the "Zoom List" screen. (☞ Page 172 Displaying a Zoom list)

■Creating an action

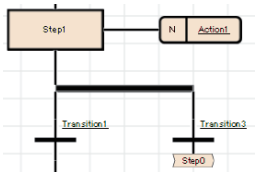
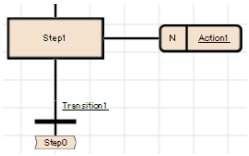
There are two ways of creation method for actions.

Each action has a different type. The type can be checked on the "Action Properties" screen.

Creation method		Operating procedure	Type
In a Zoom	Create an action with a program	☞ Page 172 Creating/displaying Zooms (action/transition)	Detailed expression
On an SFC diagram	Use ON/OFF of bit device/bit type label as an action	Select an action, then enter a bit device/bit-specified word device, or a bit type label in the action name.	Label/device

Inserting/editing jumps

■Inserting a jump

Insertion position	Method
	1. Select a transition, then select [Edit] ⇨ [Insert] ⇨ [Jump] ( ). 2. Select a step name of jump destination.
	1. Select a transition, then select [Edit] ⇨ [Modify] ⇨ [Switch between Jump Symbol and Connection Line]. 2. Select a step name of jump destination. All SFC diagram written below the inserted jump are deleted.

■Changing a jump destination

1. Select the transition which is immediately before the jump, then select [Edit] ⇨ [Modify] ⇨ [Switch between Jump Symbol and Connection Line].
   Alternatively, select the jump, and press the F2 key.
2. Select the step name of jump destination which is to be changed.

■Switching a jump/connection line

Select a jump, then select [Edit] ⇨ [Modify] ⇨ [Switch between Jump Symbol and Connection Line].

To restore a connection line to a jump, select the connection line, then select [Edit] ⇨ [Modify] ⇨ [Switch between Jump Symbol and Connection Line].

Inserting/adding selection branches and simultaneous branches

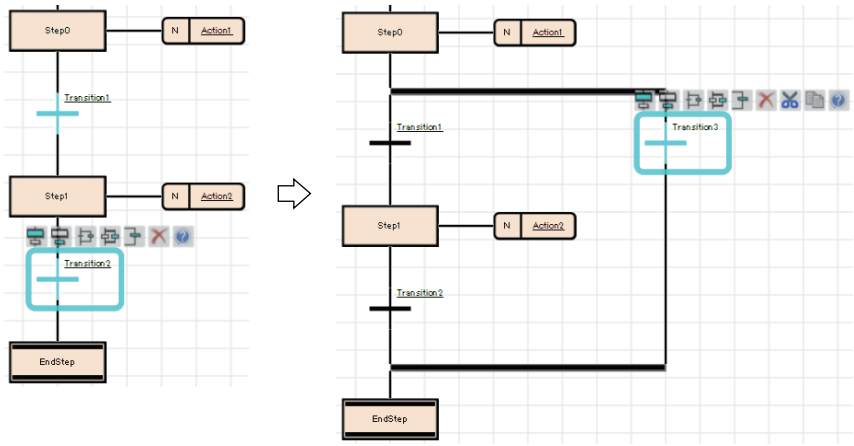
■Inserting a branch below a step/transition

Select a step/transition, then select [Edit] ⇨ [Insert] ⇨ [Selection Branch] ( )/[Simultaneous Branch] ( ).

When a selection branch/simultaneous branch was inserted, the lacking SFC elements are inserted automatically to make it be a proper SFC diagram.

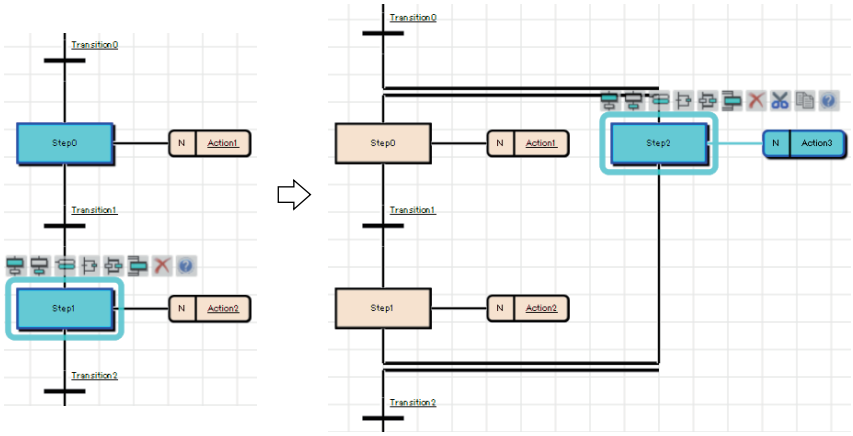
■Adding a selection branch on the right side of a transition/selection condition

Select a transition/selection branch to which a new one will be added, then select [Edit] ⇨ [Insert] ⇨ [Selection Branch Leg] ( ). (Multiple selections allowed.)



■Adding a simultaneous branch on the right side of a step/simultaneous branch

Select a step/simultaneous branch to which a new one is added, then select [Edit] ⇨ [Insert] ⇨ [Simultaneous Branch Leg] ( ). (Multiple selections allowed.)



Common operations of SFC elements

Operation		Operating procedure	
		Mouse	Keyboard
Select	To select a single element	Click an element to be selected.	Move the cursor on an element to be selected.
	To select multiple elements	• Click multiple elements with the Shift key held down. • Click the background of the SFC diagram editor, and drag it diagonally.	Select multiple elements by moving the cursor with the Shift key held down.
	To select a network	Click an element, then select [Edit] ⇨ [Select Network].	Select an element, then press the Ctrl + Shift + A keys.
	To select whole elements	The operation is same as the one for selecting multiple elements.	Press the Ctrl + A keys.

Deleting/cutting/copying/pasting/moving SFC elements

Some elements cannot be used by itself on the SFC editor. When SFC elements were deleted/cut/pasted, the other elements, that is not the target for this change, may be deleted/pasted to make a normal SFC diagram.

For the following elements, the previous elements of them also are deleted. For other elements, the next elements are deleted together.

- End step
- Step/transition that are placed immediately before a branch
- The last step/transition in a branch

A selected element cannot be pasted onto the other place where is not connected with a line on an SFC diagram.

For copying FBD/LD elements between editors, only pasting from the FBD/LD editor to the SFC diagram editor is available.

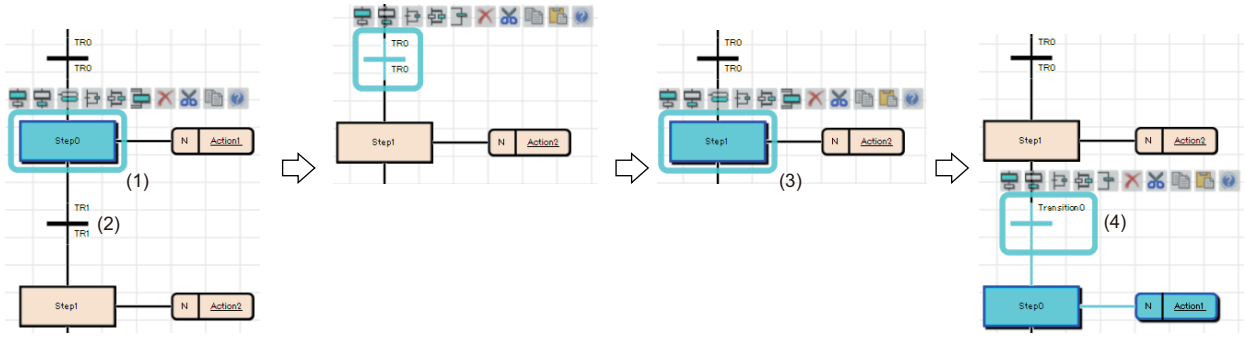
Precautions

When a Zoom has already been created, the Zoom will not be deleted if a step/transition/action is deleted on an SFC diagram. Delete it on the "Zoom List" screen. (☞ Page 172 Displaying a Zoom list)

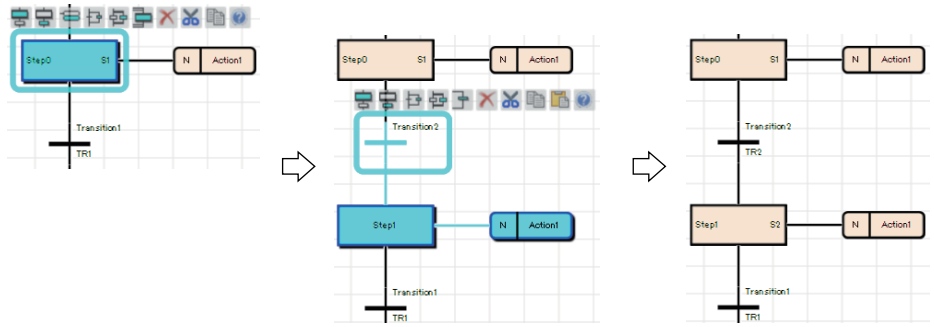
If the action that the Zoom has been created is copied, the Zoom is not copied to the destination.

■Step

- When deleting an end step, the transition placed immediately before the end step also is deleted. Thereby, the step that connects to the above transition is changed to an end step. If the element above of the end step is a branch, the branch also is deleted.
- Operation examples of cutting/pasting:
  If the step (1) was cut, the next transition (2) is deleted together. If the step is pasted with selecting the step (3), the lacking transition (4) of the step (3) will be inserted.



- Duplicate step names and Step No. cannot exist on an SFC diagram.
  When copying a step of which the step name is 'Step0' and the Step No. is 'S1' on a same SFC diagram, the step is pasted with a different name (Step1) and without the Step No. (S1). Step No. (S2) is assigned again by convention.



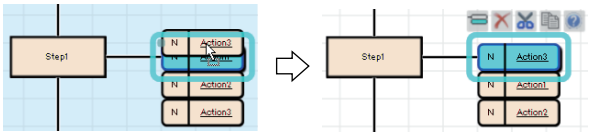
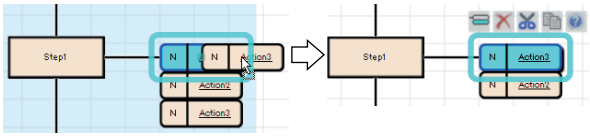
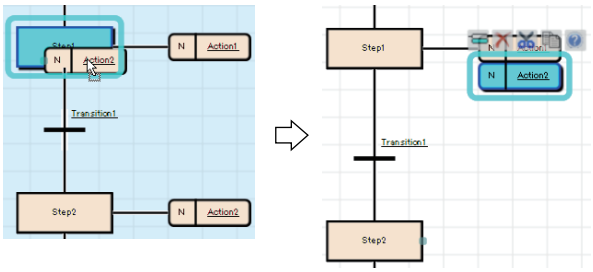
■Transition

Duplicate Transition No. cannot exist on an SFC diagram. When copying a Transition No. on a same SFC diagram, the Transition No. is pasted without Transition.No. Transition No. (TR□) is assigned again by convention.

■Action

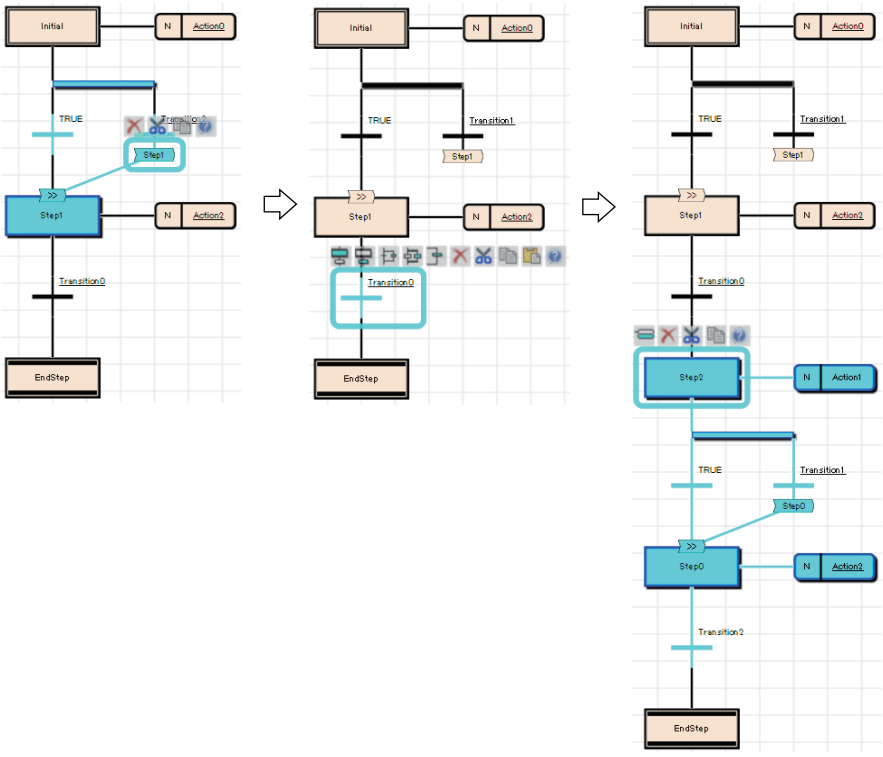
Cut/copy an action, and paste it selecting a step/action of the destination.

Actions can be copied/moved by dragging and dropping. Note that the methods are different depending on the place where it will be dropped.

Method	Example
Drag and drop an action (Action3) onto a target action (Action1). The orders of those actions will be switched.	
Drag and drop an action (Action3) to the same height of a target action (Action1). The action (Action1) will be overwritten.	
Drag and drop an action (Action2) onto a target step (Step1). The action (Action2) will be moved. (An action (Action2) can be copied by dragging the action with pressing Ctrl key.)	

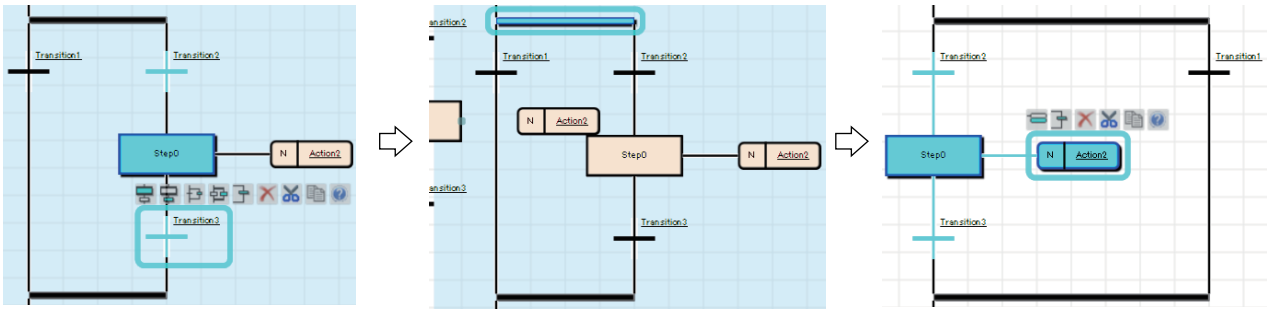
■Jump

- Deleting a jump only is not applicable. Switch the jump to a connection line first. (☞ Page 171 Jump → selection branch (connection))
- It is not applicable to copy a jump only. Copy a jump source and a jump destination, and paste them.
  The following is an example image when selecting and pasting a transition (Transition0). When selecting an end step, it is pasted as well.



■Selection branch

Drag and drop a step or a transition onto the left/right side of a selection branch. The orders of those branches is switched.

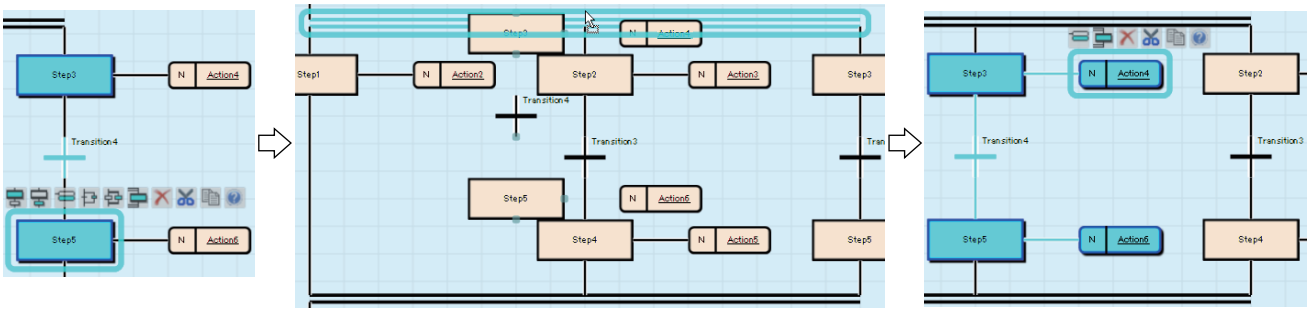


Precautions

The left side of transition is given priority to perform when both side of transitions were satisfied at a same time. Check the operation of SFC program before switching branches.

■Simultaneous branch

Drag and drop a step and a transition onto the simultaneous branch. The orders of those branch will be switched.



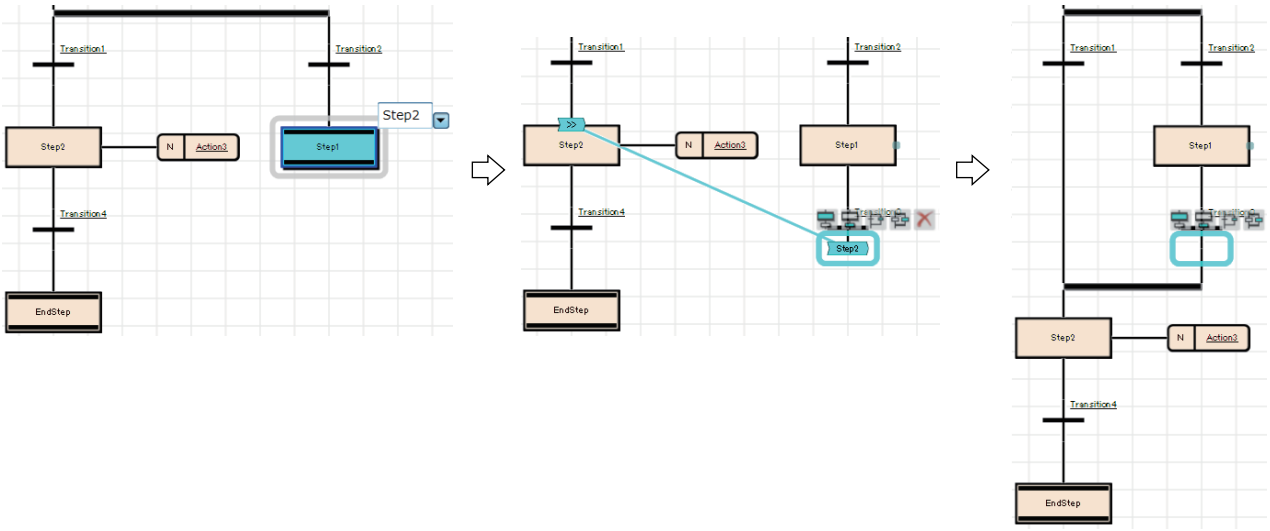
Changing SFC elements

■End step → normal step

For restoring a step which has been changed to an end step (exanple: Step1) once to a normal step, switch it to a jump, then switch the jump symbol to a connection line.

Ex.

1. Select an end step (Step1), and select [Edit] ⇨ [Modify] ⇨ [End Step/Jump]. Then select a step name (Step2) which has been placed under the branch before changing.
2. Select the added jump, then select [Edit] ⇨ [Modify] ⇨ [Switch between Jump Symbol and Connection Line].

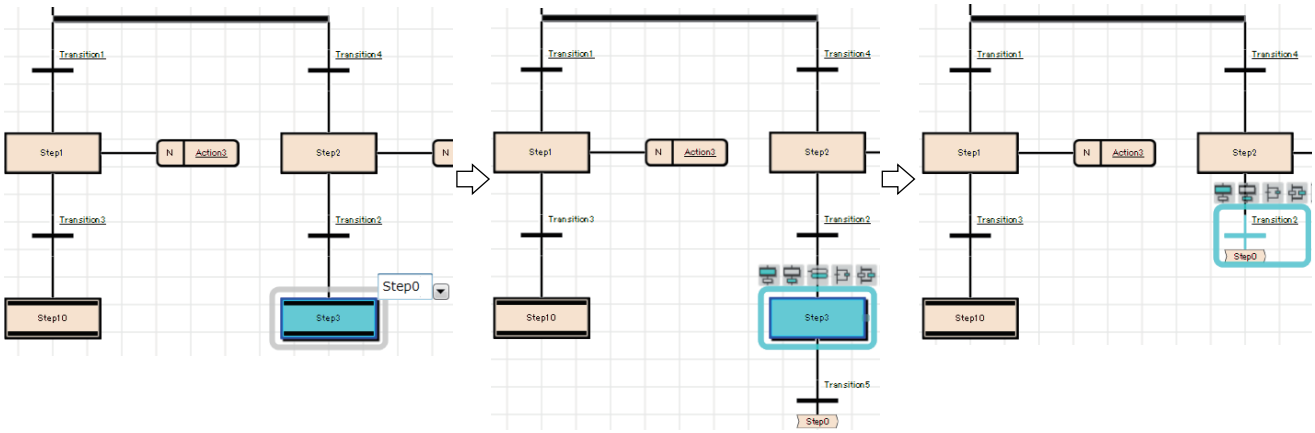


■End step → jump

After switching an end step to a jump on the selection branch, delete the unnecessary step.

Ex.

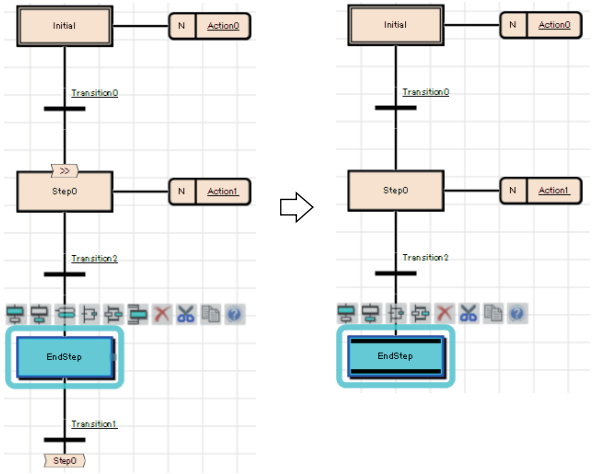
1. Select an end step to be changed (Step3), and select [Edit] ⇨ [Modify] ⇨ [End Step/Jump]. Then select a jump destination (Step0).
2. Delete the unnecessary step (Step3).



■Jump → end step

Ex.

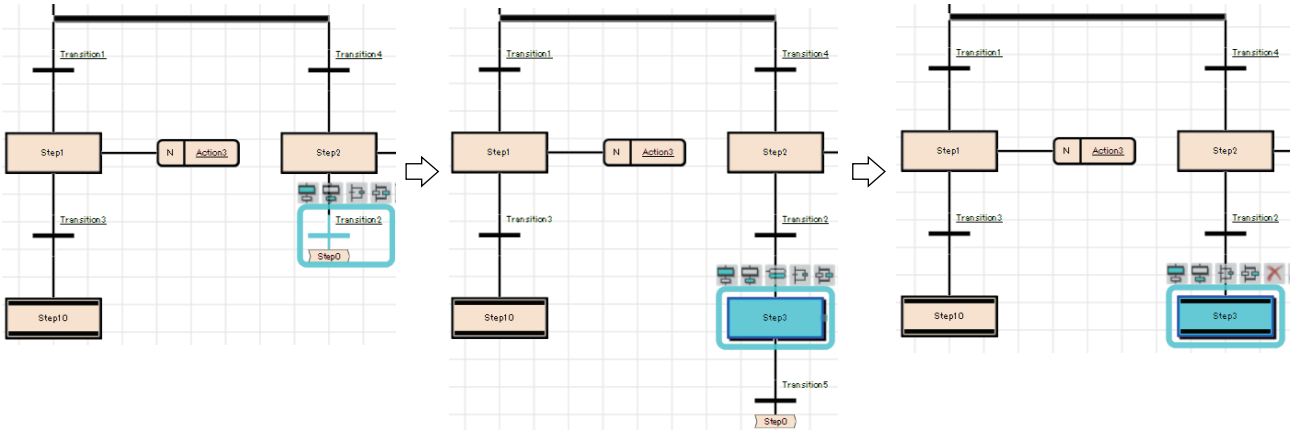
Select a step which places before a jump, then select the [Edit] ⇨ [Modify] ⇨ [End Step/Jump].



For a jump on a selection branch, insert a step before a jump to be changed, and change the step to an end step.

Ex.

1. Insert a step (Step3) before a jump to be changed.
2. Select the inserted step, then select [Edit] ⇨ [Modify] ⇨ [End Step/Jump].

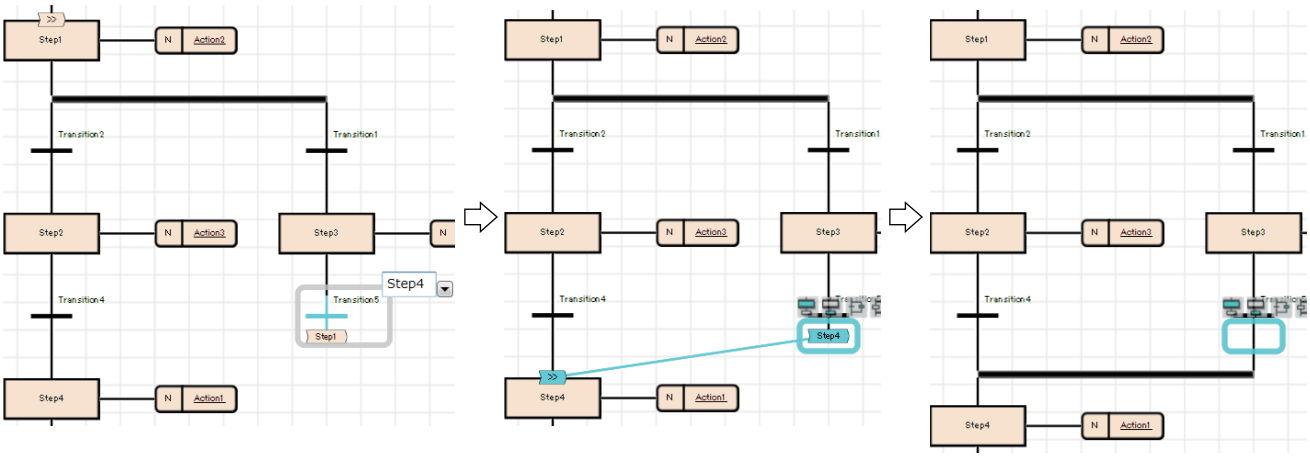


■Jump → selection branch (connection)

Change a jump destination to the step which is immediately after the branch to be connected, then switch it to a connection line.

Ex.

1. Select the transition (Transition5) that places immediately before a jump to be changed, and select [Edit] ⇨ [Modify] ⇨ [Switch between Jump Symbol and Connection Line]. Then, select the step (Step4) which will be connected with the connection line.
2. Select the jump and select [Edit] ⇨ [Modify] ⇨ [Switch between Jump Symbol and Connection Line].



Converting blocks

Check whether there is any problems on an activated SFC diagram. The program in the Zoom will not be checked.

Operating procedure

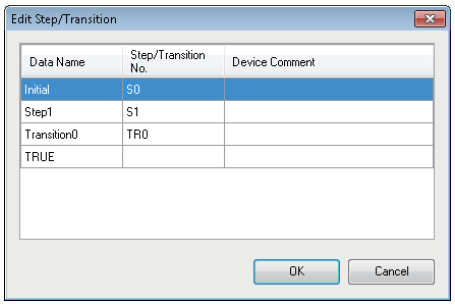
Select [Convert] ⇨ [Convert Block].

Editing Step No./Transition No.

Step No. (S☐)/Transition No. (TR☐) in a block and its device comments can be checked/changed in a list.

Window

[Edit] ⇨ [Edit Step/Transition]



Operating procedure

Select a column of device to be changed, then change the device, and click the [OK] button.

Creating/displaying Zooms (action/transition)

Programs of actions/transitions can be created in languages corresponding to contents.
The method for entering programs is same as the one for each program language.
To display the SFC diagram which is to be the start source, select [View] ⇨ [Open Zoom Source Block].

Operating procedure

1. Double-click an action name/transition name on an SFC diagram.
2. Set each item on the "New Data" screen, then click the [OK] button.

The Zoom appears when the program has already been created.

Precautions

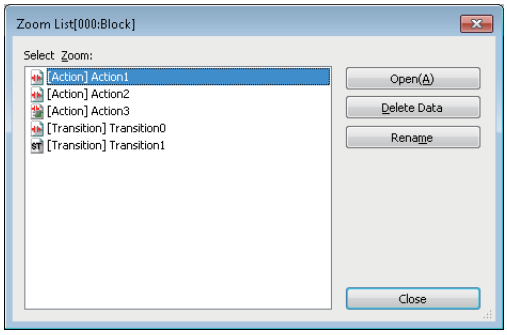
Some instructions cannot be used to actions/transitions. For details, refer to the following manual.
📖 MELSEC iQ-R Programming Manual (Program Design)

Displaying a Zoom list

Display a created Zoom in a list.
Displaying/deleting/changing name of Zoom is possible in a list.

Window

- Open the target SFC diagram editor, then select [View] ⇨ [Open Zoom List].
- Select "Program" ⇨ "(execution type)" ⇨ "(program file)" ⇨ "(block)" on the Project view, then right-click and select [Zoom List] from the shortcut menu.

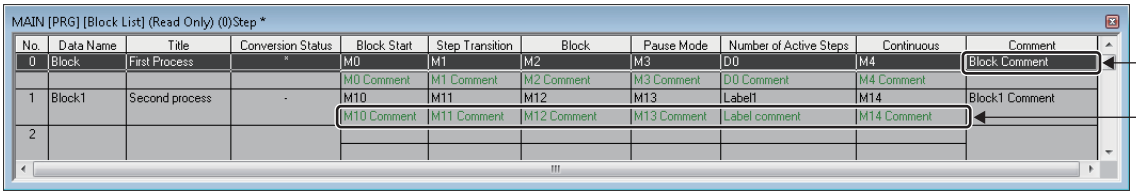


Displaying an SFC block list

Display a data name of a block, title, conversion status, and block information in a program file in a list.
For details on the block information, refer to the following manual.
📖 MELSEC iQ-R Programming Manual (Program Design)

Window

- Open the target SFC diagram editor, then select [View] ⇨ [Open SFC Block List].
- Select "Program" ⇨ "(execution type)" ⇨ "(program file)" on the Project view, then right-click and select [Open SFC Block List] from the shortcut menu.



Displayed items

Item	Description	Related operation
Comment	Comments set on the "Properties" screen of block.	■Show/Hide [View] ⇨ [SFC Block List Comment]
Device/label comment	Comments for devices/labels which were set to the block information on the "Properties" screen of block.	■Show/Hide [View] ⇨ [Device Display]

Editing/creating blocks on SFC block list

To edit existing blocks, and to create new blocks are applicable on an SFC block list.

Operating procedure

Double-click an editing target block. When creating a new block, select an empty row.

Copying blocks

Operating procedure

1. Select a block to be copied, then select [Edit] ⇨ [Copy] ( ).
2. Select a block to be pasted, then select [Edit] ⇨ [Paste] ( ).
3. Select whether to change or overwrite its block No. on the confirmation screen, then click the [OK] button.
4. Select an item to be pasted on the "Contents to Paste" screen, then click the [OK] button.

Point

Blocks copied on the Project view can be pasted on a block list.
Multiple blocks can be selected and copied when copying and pasting on the Project view.

Precautions

If "Step/Transition Comment" was selected on the "Contents to Paste" screen, steps/transition comments in the device comment are pasted after pasting all copied data. Thereby, steps/transition comments may not be pasted when the [Cancel] button was clicked during processing even if data pasting has been already completed.

Search

■Searching for block information

Search for block information (device/label) on an SFC block list.

Operating procedure

1. Select [Find/Replace] ⇨ [Block Information Find Device].
2. Enter a device/label to be searched, then click the [Find Next] button.

■Jump

Specify a block No. or a block name, and move the cursor on an SFC block list.

Operating procedure

1. Select [Find/Replace] ⇨ [Jump].
2. Select a block No./block name on the "Jump" screen, then click the [OK] button.

Point

The "Jump" screen can be displayed by pressing a numeric key of the keyboard on an SFC block list.

Displaying SFC diagrams

Display an SFC diagram of block, which is pointed with the cursor, from an SFC block list.

Operating procedure

1. Move the cursor on a block to be displayed.
2. Select [View] ⇨ [Open SFC Body], or double-click a block to be displayed.

Displaying local label editor

Display a local label editor of block, which is pointed with the cursor, from an SFC block list.

Operating procedure

1. Move the cursor on a block to be displayed.
2. Select [View] ⇨ [Open Label Setting].

Searching for/replacing programs

The search functions that can be used in the SFC diagram editor as follows:

Function name	Description	Reference
Cross reference	Checks devices, label declaration locations, and referenced locations in a list.	Page 200 SEARCHING DATA
Device list	Checks the usage of devices being used.	
Search and replace	• Searches for and replaces a device name, label name, instruction name, and character strings. • Changes open/close contact. • Replaces devices in a batch.	

Precautions

- When a Zoom has been created, the data name of Zoom will not be changed if the character strings of transition name/action name are replaced on an SFC diagram. Change them on the "Zoom List" screen. (☞ Page 172 Displaying a Zoom list)
- Block information is not searched in the cross reference, device list, search and replace. Use "Block Information Find Device" (☞ Page 174 Searching for block information) when searching for block information.

Displaying the Help

Check the elements used in the SFC program on the e-Manual Viewer.

To check, files of the corresponding programming manuals requires to be registered in e-Manual Viewer.

Operating procedure

1. Select a check target element.
2. Press the F1 key.

6.7 Registering Device Comments

This section explains the features and setting methods for device comments.

Device comments

There are two types of device comments; Common device comment and device comment for each program ('Each program device comment').

GX Works3 supports the global device comment/local device comment of GX Works2, and the common comment/comment by program of GX Developer.

Up to 16 comments can be set for one device. (☞ Page 49 Setting Comment Display)

Set the comments in Japanese, English, and Chinese and apply then by switching the language.

■ Difference between device comment and label comment

Device comment is a comment data to be added to devices, and read from/write to CPU module.

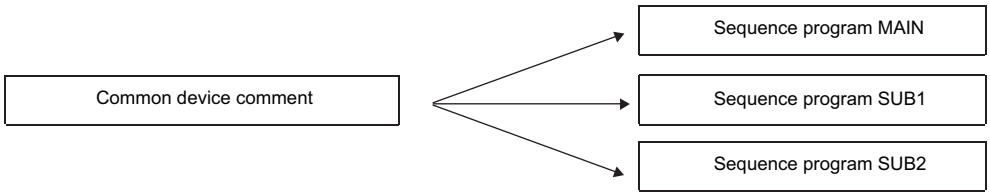
Label comment is a comment to be added to the defined labels.

Common device comment

Common device comment is a device comment that is created automatically when creating a new project.

Use this comment data when using device comments commonly in multiple programs.

The common device comments can be used if multiple programs do not exist.

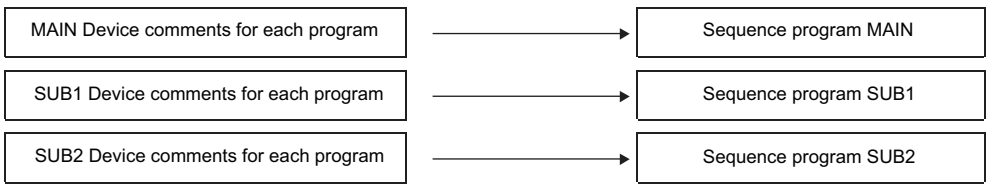


Each program device comment

'Each program device comment' is a device comment that can be created arbitrarily.

Create the device comments under the same data name as the program, and use them linked with the program.

Create a new each program device comment if necessary. (☞ Page 72 Creating projects)


MAIN Device comments for each program
SUB1 Device comments for each program
SUB2 Device comments for each program
Sequence program MAIN
Sequence program SUB1
Sequence program SUB2

'Each program device comment' can be created under the different data name from the sequence program. However, the comment will not be linked with the sequence program.

To make the comments relate to the sequence program, the comments should have the same data names as the program name.

Precautions

For FX5CPU, 'each program device comment' cannot be written to the CPU module.

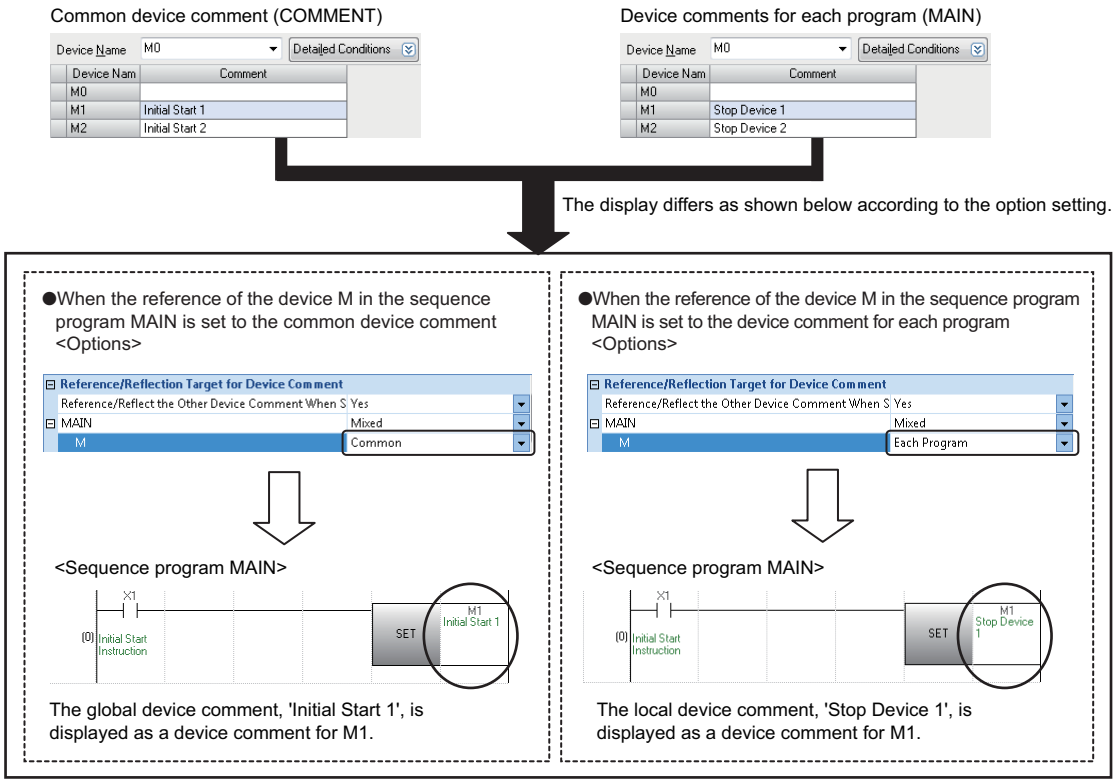
■Setting device comments for each program

When comments are set both in Common Device Comment and in Each Program Device Comment, specify the comment to be displayed on the program editor by setting the following option.

• [Tool] ⇨ [Options] ⇨ "Project" ⇨ "Device Comment Reference/Reflection Target"

Ex.

When different device comments are set for the same device (M)


Common device comment (COMMENT)
Device Name M0 Detailed Conditions
Device Nam Comment
M0
M1 Initial Start 1
M2 Initial Start 2
Device comments for each program (MAIN)
Device Name M0 Detailed Conditions
Device Nam Comment
M0
M1 Stop Device 1
M2 Stop Device 2
The display differs as shown below according to the option setting.
●When the reference of the device M in the sequence program MAIN is set to the common device comment <Options>
Reference/Reflection Target for Device Comment
Reference/Reflect the Other Device Comment When S Yes
MAIN Mixed
M Common
<Sequence program MAIN>
X1
(0) Initial Start Instruction
SET M1 Initial Start 1
The global device comment, 'Initial Start 1', is displayed as a device comment for M1.
●When the reference of the device M in the sequence program MAIN is set to the device comment for each program <Options>
Reference/Reflection Target for Device Comment
Reference/Reflect the Other Device Comment When S Yes
MAIN Mixed
M Each Program
<Sequence program MAIN>
X1
(0) Initial Start Instruction
SET M1 Stop Device 1
The local device comment, 'Stop Device 1', is displayed as a device comment for M1.

If a comment does not exist in the device comment set for "Device Comment Reference/Reflection Target", the other comment is displayed.

Applicable devices

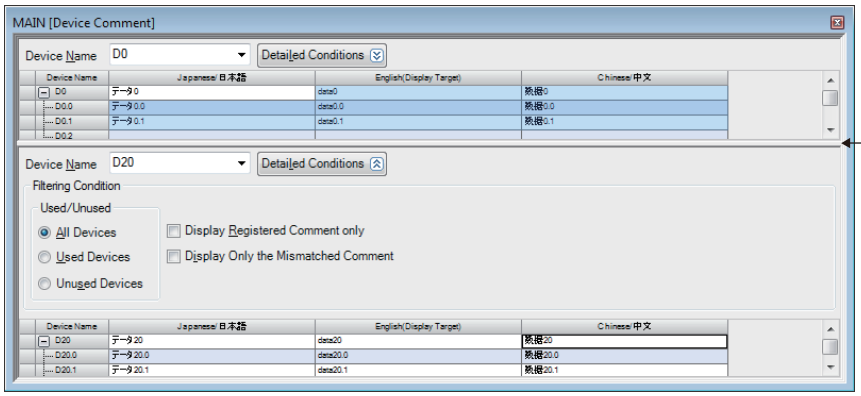
For the device types and comment setting availability, refer to the following section.

☞ Page 337 Applicable Devices in GX Works3

Configuration of device comment editor

Window

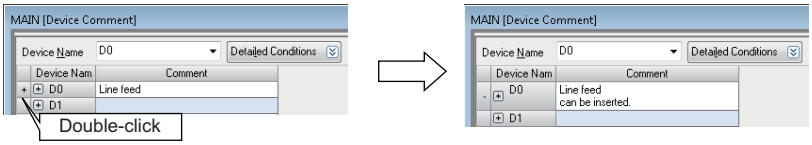
- For common device comment: Project view ⇨ "Device" ⇨"Device Comment" ⇨ "Common Device Comment"
- For device comment for each program: Project view ⇨ "Device" ⇨"Device Comment" ⇨ "Each Program Device Comment" ⇨ "(data name)"



Pressing the Ctrl + Enter keys creates a line break in a comment column.

Point

- The hidden comment is displayed by double-clicking + in the device name column.



- Fonts can be changed. (☞ Page 48 Checking and Changing Colors and Fonts)

Creating device comments

Create device comments for each device on the device comment editor.
Adding device comments make it easy to understand the content of program processing.
When less number of characters are entered, the data size to be written to CPU module can be reduced.
Set the applicable number of characters in the following option.

- [Tool] ⇨ [Options] ⇨ "Other Editor" ⇨ "Device Comment Editor" ⇨ "Number of Device Comment Editing/Displaying Characters"

When creating device comments for each program, create a device comment data in advance. (☞ Page 72 Creating projects)
The comments can be entered on a ladder editor. For details, refer to the following section.
☞ Page 135 Entering/editing comments

Operating procedure

1. Enter the device of which comment is to be set in "Device Name".
2. Enter a comment for "Comment".

Point

- By selecting two or more consecutive cells in which the comment with number is inserted, and dragging the '+' symbol displayed at the lower right corner of the cell, the data of which number is incremented can be entered.
- When cutting/copying the hidden bit-specified word device comments, select the range and [Edit] ⇨ [Cut the Range Including Hidden Bit Specification Information]/[Copy the Range Including Hidden Bit Specification Information].
  The bit-specified comments can also be pasted regardless of the setting of display/hidden by selecting [Edit] ⇨ [Paste the Range Including Hidden Bit Specification Information].

Creating multiple comments and setting display target

For details, refer to the following section.
☞ Page 49 Setting Comment Display

Detecting devices with empty cell

When comments are set in the multiple rows, detect the devices with empty cells.

Operating procedure

■Detect comments from device data being displayed on the device comment editor

Select "Display Only the Mismatched Comment" in the filter condition on the device comment editor.

■Detect comments from all device data in the device comment data

Select [Edit] ⇨ [Detect the Mismatched Comment].
Double-click "NG" on the displayed "Detect the Matched Comment of All Devices" screen to edit the device comment editor.

Deleting unused device comments

Delete unused comments in a program in a batch. To delete the comments, open the device comment editor.

Operating procedure

Select [Edit] ⇨ [Delete Unused Device Comment].

Deletion targets

Device comment type	Deletion target
Common device comment	Unused device comments in all programs
Each program device comment	Unused device comments in the corresponding program
	Device comments in all programs

The comments specified to bit-specified word device are not deleted.

Clearing all device comments

Delete the comments set to device comment data in a batch.

Operating procedure

■Delete all device data in the device comment data

Select [Edit] ⇨ [Clear All (All Devices)].

■Delete device data being displayed on the device comment editor

Select [Edit] ⇨ [Clear All (Displayed Devices)].

Importing to/exporting from files

Export/import device comments from/to file.
When multiple comments are set, the comments selected in "Available" on the "Multiple Comments Display Setting" screen are output.

Operating procedure

1. Open the device comment editor, then select [Edit] ⇨ [Import File]/[Export to File].
   In another way, select data of device comment on the Project view, then right-click and select [Import File]/[Export to File] from the shortcut menu.
2. Click the [Extended Setting] button.
3. Set the items on the "Device Comment Extended Setting" screen, and click the [OK] button.
4. Click the [Yes] button.

Precautions

When a device comment that exceeds the applicable number of characters is set on the device comment editor, a warning message appears on the Output window and excessive characters are deleted.

Formatting CSV files

The column headers of device comment editor and header names of a CSV file are linked.

- On the device comment editor, only the data of which column titles matches with header names in the CSV file is imported.
- Data can be imported even when the sorted order of CSV file does not match with the sorted order of the device comment editor.
- When importing a CSV file that is exported from GX Works3 with different language, edit the header title name in the CSV file to match the header title in the device comment editor of the import target GX Works3.

Point

The CSV files exported with GX Works2 can be imported in GX Works3.
When multiple comment display setting is set, import the CSV file after editing the header names in the CSV file exported from GX Works2 to match the header titles in the device comment editor of GX Works3.

Searching device comments

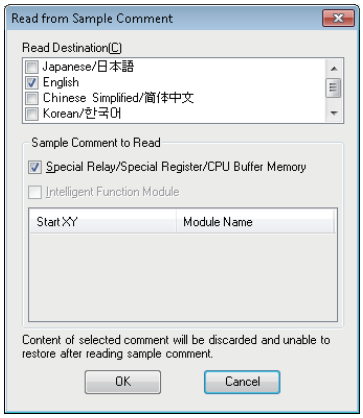
When searching device comments, refer to the following section.
☞ Page 205 Searching/Replacing Character Strings

Reading sample comments

Set the sample comments of special relays/special registers/CPU buffer memory and Intelligent function modules automatically.
Open the device comment editor in advance.
FX5CPU does not support sample comments of CPU buffer memory.

Window

[Edit] ⇨ [Read from Sample Comment]



Precautions

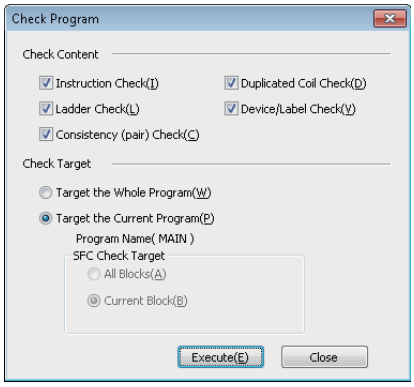
When reading sample comments of Intelligent function module, the data size may exceed the memory capacity of the CPU module and data may not be written to a CPU module. In this case, prepare an SD memory card.
For FX5CPU, the writable capacity is the same as that of the CPU module even when data is written to an SD memory card.

6.8 Checking Programs

This section explains how to check if improper input or inconsistency of programs exists.
FX5CPU does not support this function.

Window

[Tool] ⇨ [Check Program]



When the check result has an error, the error message is displayed on the Output window. Take corrective actions according to the displayed contents.

Check targets for duplicated coil

■Target instruction

The following instructions are checked as targets.

- EGP, EGF, OUT, OUTH, SET, PLS, PLF, FF, DELTA, DELTAP, SFT, SFTP, MC, Pointer, BLKMOVB*1, MOVB*1, CMLB*1

*1 When the device is T/C/ST/TL/LC/LST, it is excluded from the check.

■Target device list

The following devices are checked as targets.
The devices assigned to labels are excluded from the check.

Check range	Type	Device
Check in a project	Device	M, SM, L, F, V, S, TR, X, Y, B, SB, DX, DY, D, SD, R, ZR, RD, W, SW, T(TC), T(TS), T(TN), C(CC), C(CS), C(CN), ST(STC/SC), ST(STS/SS), ST(STN/SN), LT(LTC), LT(LTS), LT(LTN), LC(LCC), LC(LCS), LC(LCN), LST(LSTC/LSC), LST(LSTS/LSS), LST(LSTN/L), P, I, BL
	Link direct device	J□\X, J□\Y, J□\B, J□\SB, J□\W, J□\SW
	Module access device	U□\G, U□\HG
	Safety device	SA\M, SA\SM, SA\X, SA\Y, SA\B, SA\D, SA\SD, SA\W, SA\T(TC), SA\T(TS), SA\T(TN), SA\C(CC), SA\C(CS), SA\C(CN), SA\ST(STC/SC), SA\ST(STS/SS), SA\ST(STN/SN)
	Step No./ Transition No. in an SFC program	BL□\S
Check in file only	Device	#M, #V, #D, #T(TC), #T(TS), #T(TN), #C(CC), #C(CS), #C(CN), #ST(STC/SC), #ST(STS/SS), #ST(STN/SN), #LT(LTC), #LT(LTS), #LT(LTN), #LC(LCC), #LC(LCS), #LC(LCN), #LST(LSTC/LSC), #LST(LSTS/LSS), #LST(LSTN/LSN), #P
	Safety device	SA\#M, SA\#D, SA\#T(TC), SA\#T(TS), SA\#T(TN), SA\#C(CC), SA\#C(CS), SA\#C(CN), SA\#ST(STC/SC), SA\#ST(STS/SS), SA\#ST(STN/SN)

6.9 Converting Programs

This section explains how to convert a created program into a code that can be executed.
The unconverted program name is displayed red in the Project view.

Converting/converting all programs

Convert the unconverted programs in the project. Check the program at the same time.
The differences in operation between "Convert" and "Rebuild All" are shown below:

Item	Conversion target	Label assignment target	Checking program
Convert	Newly added and changed location of the program	Newly added and changed labels	Whether or not to perform the Check Program function can be specified in "Basic Setting" by selecting [Tool] ⇨ [Options] ⇨ "Convert".
Rebuild All	All the programs in the project regardless of the change	Whether to retain or reassign the labels can be selected on the "Rebuild All" screen. However, newly added labels, or labels of which data types are changed will be reassigned even when "Retain" is selected.	Whether or not to perform the Check Program function can be specified on the "Rebuild All" screen.

Operating procedure

1. Select [Convert] ⇨ [Convert] ( )/[Rebuild All] ( ).
2. For converting all programs, specify the respective items for "Label Assignment" and "Check Program" on the "Rebuild All" screen, and click the [OK] button.

Point

The unconverted data can be displayed on the Project view by selecting global label/POU/program file, and right-clicking it and selecting [Expanded/Collapse Tree] ⇨ [Open Unconverted Data] from the shortcut menu to display the status of unconverted data.

Considerations for conversion

■Canceling conversion

The conversion is canceled in a program file unit. Therefore, even when the [Cancel] button is clicked during the conversion, the conversion will not be canceled until the conversion for one program file is completed.
Note that a large-scale program file with multiple FBs/FUNs requires a longer time to cancel the conversion.

■When creating a sequence program with multiple instructions

'LD SM400' may be added in the beginning of the program in order to execute the program.

Operations that requires conversion for all programs

All programs need to be converted when one of the following operations is performed.

- Change of system parameter (Fixed Scan Communication Area Setting)
- Changing CPU parameters (File Register Setting, Device/Label Memory Area Setting, Index Register Setting, Refresh Memory Setting, Pointer Setting, Program Setting, FB/FUN File Setting, Refresh Setting (At I45 Exe))
- Changing option settings ("Data Length of Character String Data Type" in "Label Editor Common", "Convert")
- Changing module type

Considerations after converting (reassigning) all programs

When all programs are converted (reassigned), the label memory area is reassigned to all POUs. If all the converted programs are written to the CPU module and RUN as they are, the programs may be processed with the device values set before the program change.

After converting all programs, initialize the labels by the following procedure.

Operating procedure

1. Switch the CPU module to STOP.
2. Select "Device/Label Memory" on the "Memory Management" screen displayed by selecting [Online] ⇨ [CPU Memory Operation] to clear the values.
3. Select [Online] ⇨ [Write to PLC] ( ) to write the changed program files. When the initial values are set to the labels used in the program, write the label initial value file as well.
4. Reset the CPU module.

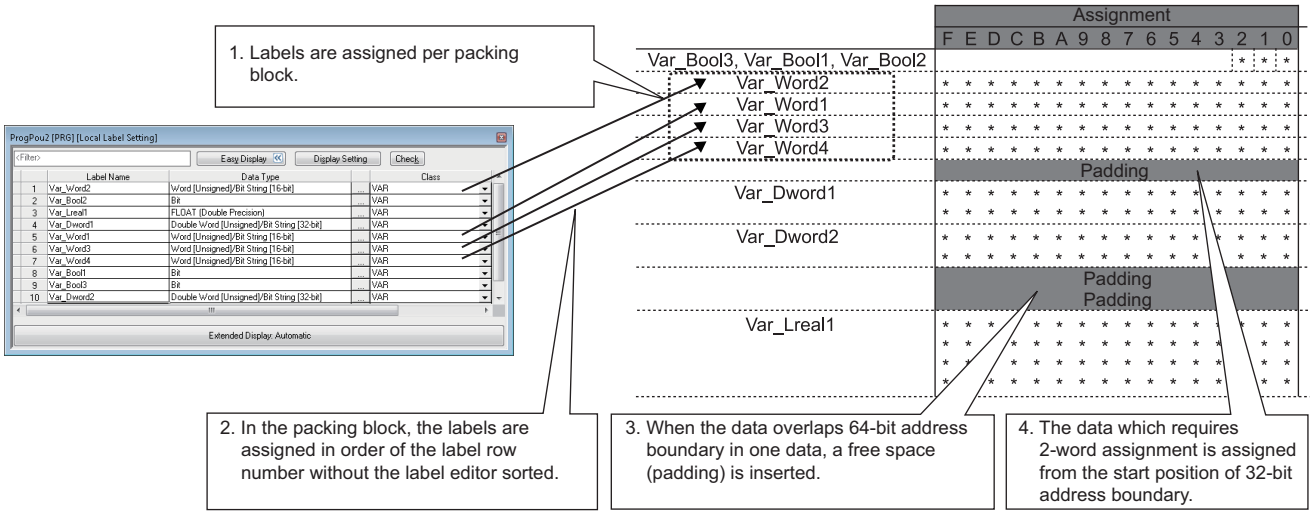
It can also be reset by selecting [Online] ⇨ [Remote Operation].

The labels are reset to '0' or initialized with the values set to the label initial value file.

Memory assignment of labels

When converting/converting all programs, the labels declared in the label editor are assigned to the memory in a packing block (which is classified as type and data type, and set them in consecutive order of line numbers) unit. Since it is assigned in POU units, the start position of POU will be 64-bit address boundary.

Ex.



The types of packing blocks and assignment order is shown below.

They are arranged in ascending order due to minimize the useless space.

Assignment order	Packing block	Data type	Remarks
1	Bit	Bit	☞ Page 186 Assignment of bit type
2	Word	Word [Unsigned]/Bit String [16-bit] Word [Signed]	—
3	String	String	Padding is not inserted because a packing block is assigned per one word unit, though the assignment amount differs depending on the number of character strings.
4	String [Unicode]	String [Unicode]	
5	Double Word	Double Word [Unsigned]/Bit String [32-bit] Double Word [Signed]	—
6	Time	Time	—
7	FLOAT [Single Precision]	FLOAT [Single Precision]	—

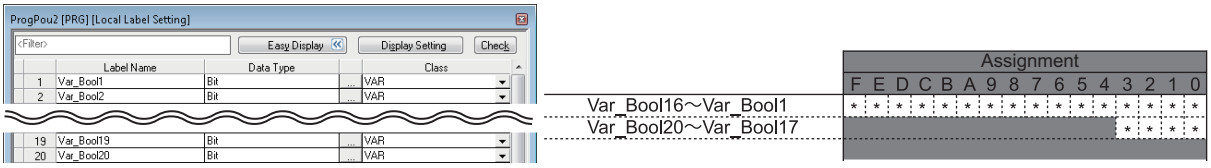
Assignment order	Packing block	Data type	Remarks
8	Timer	Timer	☞ Page 187 Assignment of timer/retentive timer/counter
9	Retentive Timer	Retentive Timer	
10	Counter	Counter	
11	FLOAT [Double Precision]	FLOAT [Double Precision]	—
12	Long Timer	Long Timer	☞ Page 187 Assignment of timer/retentive timer/counter
13	Long Retentive Timer	Long Retentive Timer	
14	Long Counter	Long Counter	
15	Array	All	☞ Page 186 Assignment of array
16	Structure/FB instance	All	☞ Page 187 Assignment of structure

■Assignment of bit type

If the number of bit type labels exceeds 16, the labels will be assigned to the next memory area consecutively.

When using the label of which the class is "VAR_INPUT"/"VAR_OUTPUT" in the function block, the area of EN/ENO is assigned to the head of FB instance.

Ex.



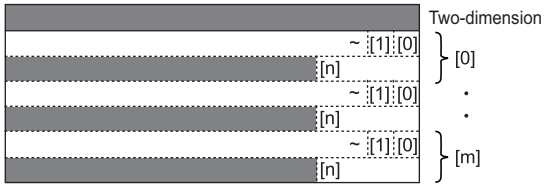
■Assignment of array

For a bit type array, the memory is assigned from the start address (0 bit) of the array, and continuous bits for one-dimensional element are assigned in word unit. As for the two-dimensional element or later, the same area as the one-dimensional element is assigned for the number of the dimensional elements continuously.

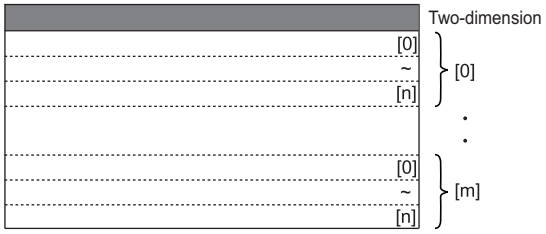
For an array other than bit type, the area which is required for the element is assigned for the memory of the data type continuously. A padding is not inserted between the array elements.

Ex.

<Assignment example of bit type two-dimension array_Bit (0..n, 0..m)>



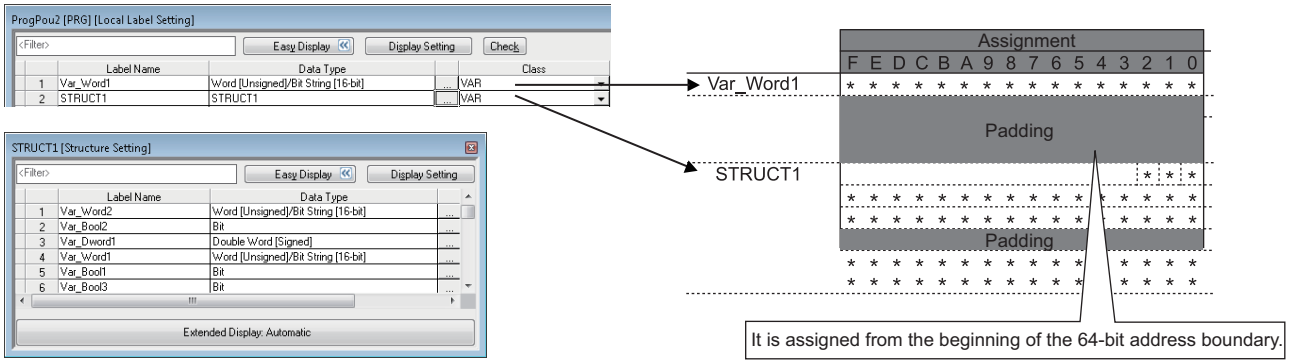
<Assignment example of word type two-dimension array_Word (0..n, 0..m)>



■Assignment of structure

A member of structure is assigned according to the assignment order of packing blocks. When a structure is declared in the member of structure, it will be assigned in a packing block according to the order noted previously. It will be assigned from the start position of 64-bit address boundary.

Ex.



■Assignment of timer/retentive timer/counter

● Timer, retentive timer, counter

Timer type, retentive timer type, and counter type are the maker-defined structures that have the same data type as MELSEC 16-bit timer device (T), bit retentive timer device (ST), and bit counter device(C). The member configurations of each data type are as follows:

Type	Member	Data type	Description
Timer	S	Bit type	The same operation as the contact (TS) of the timer device.
	C	Bit type	The same operation as the coil (TC) of the timer device.
	N	Word [Unsigned]/Bit String [16-bit]	The same operation as the current value (TN) of the timer device.
Retentive timer	S	Bit type	The same operation as the contact (STS) of the retentive timer device.
	C	Bit type	The same operation as the coil (STC) of the retentive timer device.
	N	Word [Unsigned]/Bit String [16-bit]	The same operation as the current value (STN) of the retentive timer device.
Counter	S	Bit type	The same operation as the contact (CS) of the counter device.
	C	Bit type	The same operation as the coil (CC) of the counter device.
	N	Word [Unsigned]/Bit String [16-bit]	The same operation as the current value (CN) of the counter device.

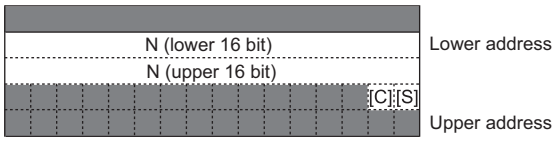
Each of timer, retentive timer, and counter requires 2 words.

● Long timer, long retentive timer, long counter

Long timer type, long retentive timer type, and long counter type are the maker-defined structures that have the same data type as MELSEC 32-bit timer device (LT), bit retentive timer device (LST), and bit counter device(LC). The member configurations of each data type are as follows:

Type	Member	Data type	Description
Long timer	S	Bit type	The same operation as the contact (LTS) of the timer device.
	C	Bit type	The same operation as the coil (LTC) of the timer device.
	N	Double Word [Unsigned]/Bit String [32-bit]	The same operation as the current value (LTN) of the timer device.
Long retentive timer	S	Bit type	The same operation as the contact (LSTS) of the retentive timer device.
	C	Bit type	The same operation as the coil (LSTC) of the retentive timer device.
	N	Double Word [Unsigned]/Bit String [32-bit]	The same operation as the current value (LSTN) of the retentive timer device.
Long counter	S	Bit type	The same operation as the contact (LCS) of the counter device.
	C	Bit type	The same operation as the coil (LCC) of the counter device.
	N	Double Word [Unsigned]/Bit String [32-bit]	The same operation as the current value (LCN) of the counter device.

Each of long timer, long retentive timer, and long counter requires 4 words.



Converting character codes

Character strings are handled as Unicode in GX Works3 and they are handled as ASCII in a CPU module. When a converting program, character strings also are converted from Unicode to ASCII. When the reading program from a programmable controller, the character strings are converted from ASCII to Unicode.

Performing online program change simultaneously with conversion

For the operation methods for performing the online program change function simultaneously with conversion, refer to the following section.

☞ Page 258 Writing Programs While CPU is in RUN

Checking for errors and warnings

When the program is converted, the target programs and label settings are checked and the result is displayed on the Output window.

The corresponding error location can be referred on the Output window.

Operating procedure

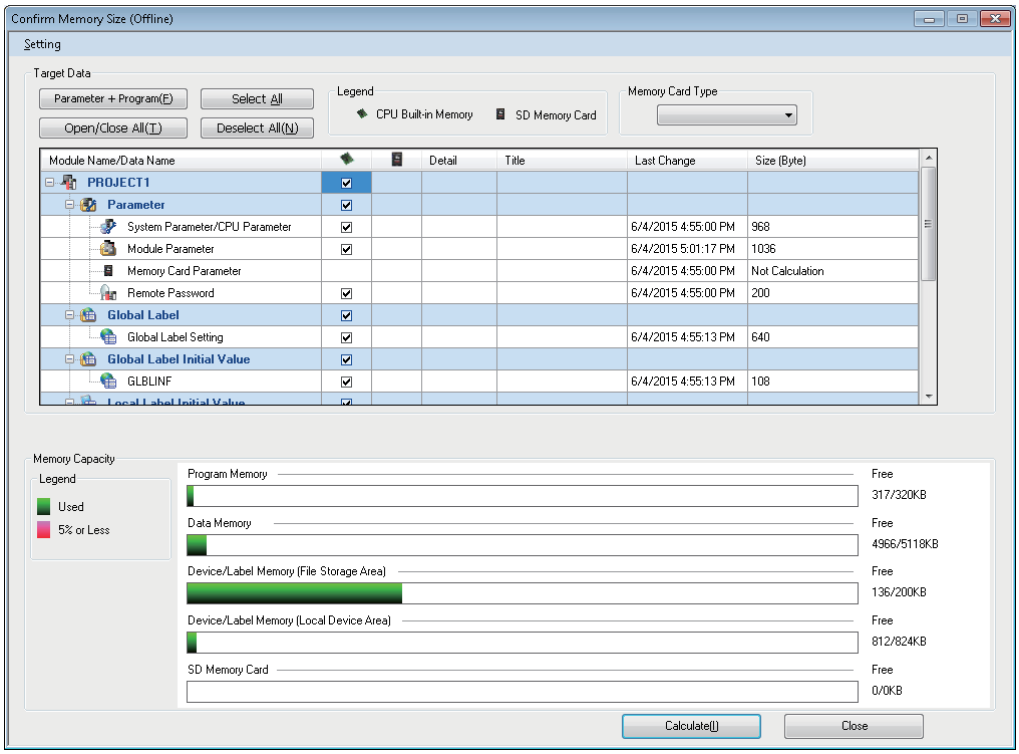
1. Double-click the error/warning message displayed on the Output window.
2. Check the corresponding error location and modify the error as instructed by the error/warning message.

6.10 Calculating Memory Size

This section shows how to calculate offline a memory size required when writing data in a project to a CPU module .

Window

[Tool] ⇨ [Confirm Memory Size (Offline)]



Operating procedure

1. When the data to be written to an SD memory card is included, select the corresponding SD memory card from the pull-down list of "Memory Card Type".
2. Select the file to be written, then click the [Calculate] button.

The methods of the detailed setting for a target file or each data are same as the one on the "Online Data Operation" screen. Refer to the following section.

☞ Page 248 WRITING/READING DATA TO CPU MODULE

7 SETTING DEVICE MEMORY

This chapter explains the features and settings of the device memory.

7.1 About Device Memory

The device memory managed by GX Works3 is a data that reads/writes values from/to the device memory on a CPU module. By reading device memory, the state of device memory in a CPU module can be checked. It can be used for offline debugging.

Additionally, the current values of device memory in a CPU module can be changed in a batch by writing device memory to a CPU module.

Read/write device memory on the "Online Data Operation" screen.

For details, refer to the following section.

☞ Page 248 Writing/Reading Programmable Controller Data

Applicable devices

For the devices that can be set on the Device Memory editor and its input method, refer to the following section.

☞ Page 337 Applicable Devices in GX Works3

Writing device memory to global devices/local devices

Device memory does not distinguish the global devices and the local devices.

Write device memory to each device in a CPU module by each of the following methods.

FX5CPU does not support local devices.

- Global device: Create device memory data with a different name from the one for the program file, and write it to a CPU module
- Local device: Create device memory with a same name as the one for the program file, and write it to a CPU module

7.2 Configuration of Device Memory Editor

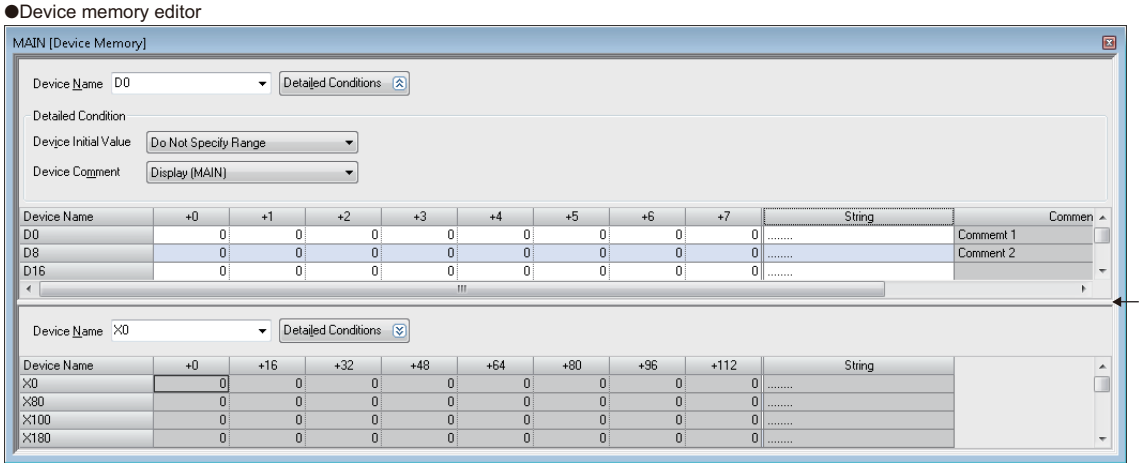
This section explains the screen configuration of the Device Memory editor.

Window

Project view ⇨"Device" ⇨"Device Memory" ⇨ "(data name)"

●Toolbar

●Device memory editor


MAIN [Device Memory]
Device Name D0
Detailed Conditions
Detailed Condition
Device Initial Value
Do Not Specify Range
Device Comment
Display (MAIN)
Device Name +0 +1 +2 +3 +4 +5 +6 +7 String Commen
D0 0 0 0 0 0 0 0 0 ..... Comment 1
D8 0 0 0 0 0 0 0 0 ..... Comment 2
D16 0 0 0 0 0 0 0 0 .....
Device Name X0
Detailed Conditions
Device Name +0 +16 +32 +48 +64 +80 +96 +112 String
X0 0 0 0 0 0 0 0 0 .....
X80 0 0 0 0 0 0 0 0 .....
X100 0 0 0 0 0 0 0 0 .....
X180 0 0 0 0 0 0 0 0 .....
Separation bar

Only the range of the device initial value is displayed when selecting "Specify Range" from "Device Initial Value".

Point

Fonts can be changed.

☞ Page 48 Checking and Changing Colors and Fonts

Setting display format

Set the display format (Display Unit Format, Data Display Format, String Display Format, Value, Bit Order, Switch No. of Points) of the device value being displayed.

Operating procedure

1. Select [View] ⇨ [Display Format Detailed Setting] ( ).
2. Set the items on the "Display Format" screen, and click the [OK] button.

The setting can be performed by clicking .

7.3 Setting Device Memory

This section explains how to set the devices and device values.

Precautions

- If a deletion is executed after selecting a cell, the device value is cleared to '0'.
- When performing the copy/paste operation on the Device Memory editor, the result is as follows:
  The value is pasted in a 16-bit integer [signed] or decimal format string delimited by tab, and displayed with the format of the editor to be pasted.
- The entering method by prefixing '#' to a device name to regard it as a local device is not supported.
  Select "Global Label" or "Local Label" when reading device memory from the CPU module.

Setting device values in units of points

Set a device and device value in units of points.

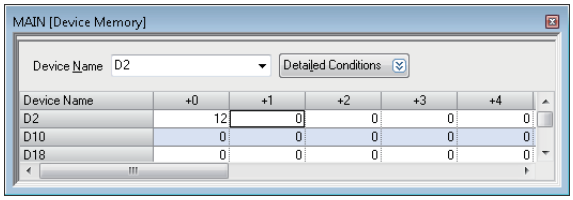
Ex.

Set the following value in a word multi-point format.

Device: D2, Device value: 12

Operating procedure

1. Set the display format. (☞ Page 192 Setting display format)
2. Enter 'D2' in "Device Name".
3. Enter '12' to the device 'D2'.



Point

When the display unit format is bit, the device value can be changed by double-clicking a cell.

Setting values in batch

Set the same value to consecutive devices.

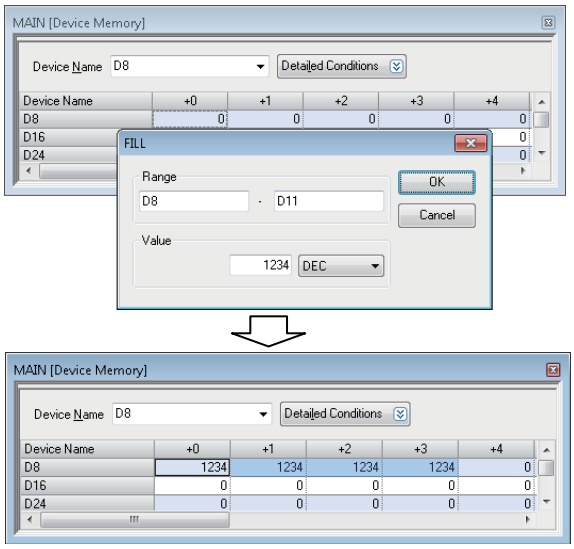
Ex.

Set the following values in a word multi-point format.

Device: D8 to D11, Device value: 1234

Operating procedure

1. Set the display format. (☞ Page 192 Setting display format)
2. Enter 'D8' to "Device Name", and select the range of the device value to be set.
3. Select [Edit] ⇨ [FILL] ( ).
4. Set the items on the "FILL" screen, and click the [OK] button.



Setting character strings

Set a character string to a device.

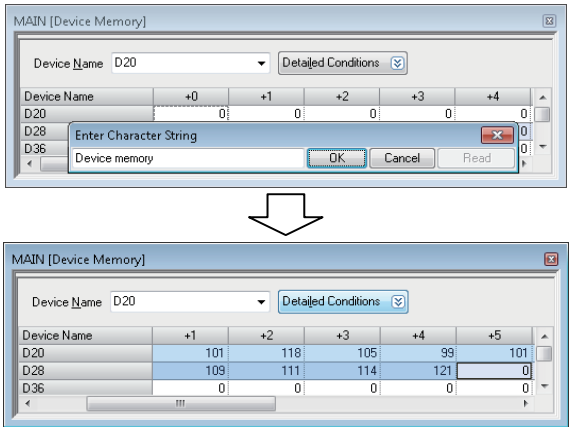
Ex.

Set the following value in a word multi-point format.

Device: D20, Character string: Device memory

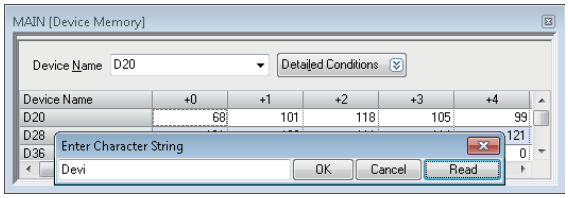
Operating procedure

1. Set the display format. (☞ Page 192 Setting display format)
2. Enter 'D20' to "Device Name", and select [Edit] ⇨ [Enter Character String].
3. Enter a character string on the "Enter Character String" screen, and click the [OK] button.



Point

- Character string can be entered to "String" column directly.
- Click the [Read] button on the "Enter Character String" screen to read the set device values (string) one by one.



Precautions

For device memory, use the characters in the Unicode Basic Multilingual Plane.

If the characters outside the Unicode Basic Multilingual Plane are specified, they may not be edited in the program.

Clearing whole memory of device memory

Clear the data set to device memory data to '0' in batch in the project.

Operating procedure

■Delete all data in the device memory data

Select [Edit] ⇨ [Clear All (All Devices)].

■Delete the devices displayed on the Device Memory editor

Select [Edit] ⇨ [Clear All (Displayed Devices)].

Interaction with device initial value

Device memory function registers/utilizes values to/from the device initial value.
Create a device initial value data to be registered in advance.

Operating procedure

1. Click the [Detailed Conditions] button.
2. Select "Specify Range" in the detailed conditions of "Device Initial Value".
3. Select [Edit] ⇨ [Register/Import Device Initial Value].

The Device Initial Value editor is displayed. For details on the Device Initial Value editor, refer to the following sections.

- Registering the value of the device memory to device initial value: ☞ Page 199 Setting device initial values
- Utilizing values from device initial values to device memory: ☞ Page 199 Registering to device memory

7.4 Writing/Reading Data to/from CPU Module

When writing/reading the value of device memory to/from the CPU module, perform them on the "Online Data operation" screen in file units.

8 SETTING DEVICE INITIAL VALUES

This chapter explains how to set the device initial values.

8.1 About Device Initial Values

The device initial value managed by GX Works3 is a data that can be read from/write to the device in a CPU module.
For details of device initial values, refer to the following manuals.
📖 MELSEC iQ-R CPU Module User's Manual (Application)
📖 MELSEC iQ-F FX5 User's Manual (Application)

Applicable devices

For devices that can be set on the Device Initial Value editor, refer to the following section.
☞ Page 337 Applicable Devices in GX Works3

Writing device initial values to global devices/local devices

Device initial value does not distinguish the global device and local device.
Write device initial values to each device in a CPU module by the following methods.
FX5CPU does not support local devices.

- Global device: Create device initial value data with a different name from the one for the program file, and write it to a CPU module
- Local device: Create device initial value data with a same name as the one for the program file, and write it to a CPU module

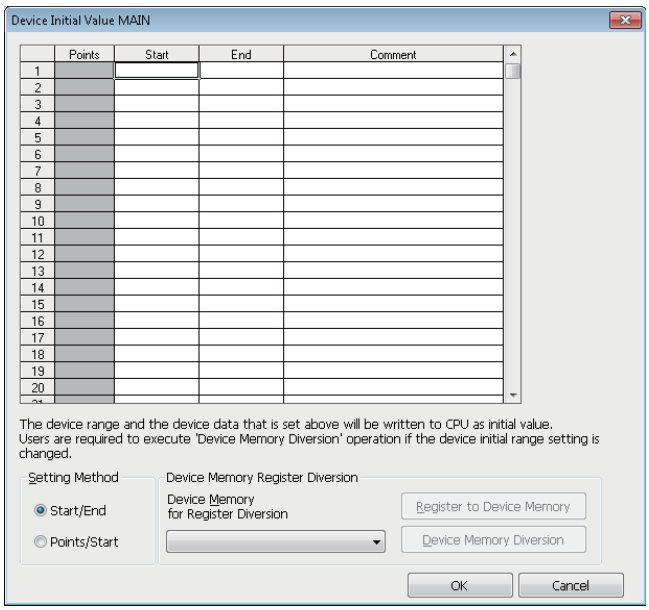
The devices which cannot be used as local devices are not written even when they are set to device initial values.

8.2 Configuration of Device Initial Value Editor

The configuration of the Device Initial Value editor is shown below.

Window

Project view ⇨ "Device" ⇨ "Device Initial Value" ⇨ "(data name)"


Device Initial Value MAIN
Points
Start
End
Comment
The device range and the device data that is set above will be written to CPU as initial value.
Users are required to execute 'Device Memory Diversion' operation if the device initial range setting is changed.
Setting Method
Start/End
Points/Start
Device Memory Register Diversion
Device Memory for Register Diversion
Register to Device Memory
Device Memory Diversion
OK
Cancel

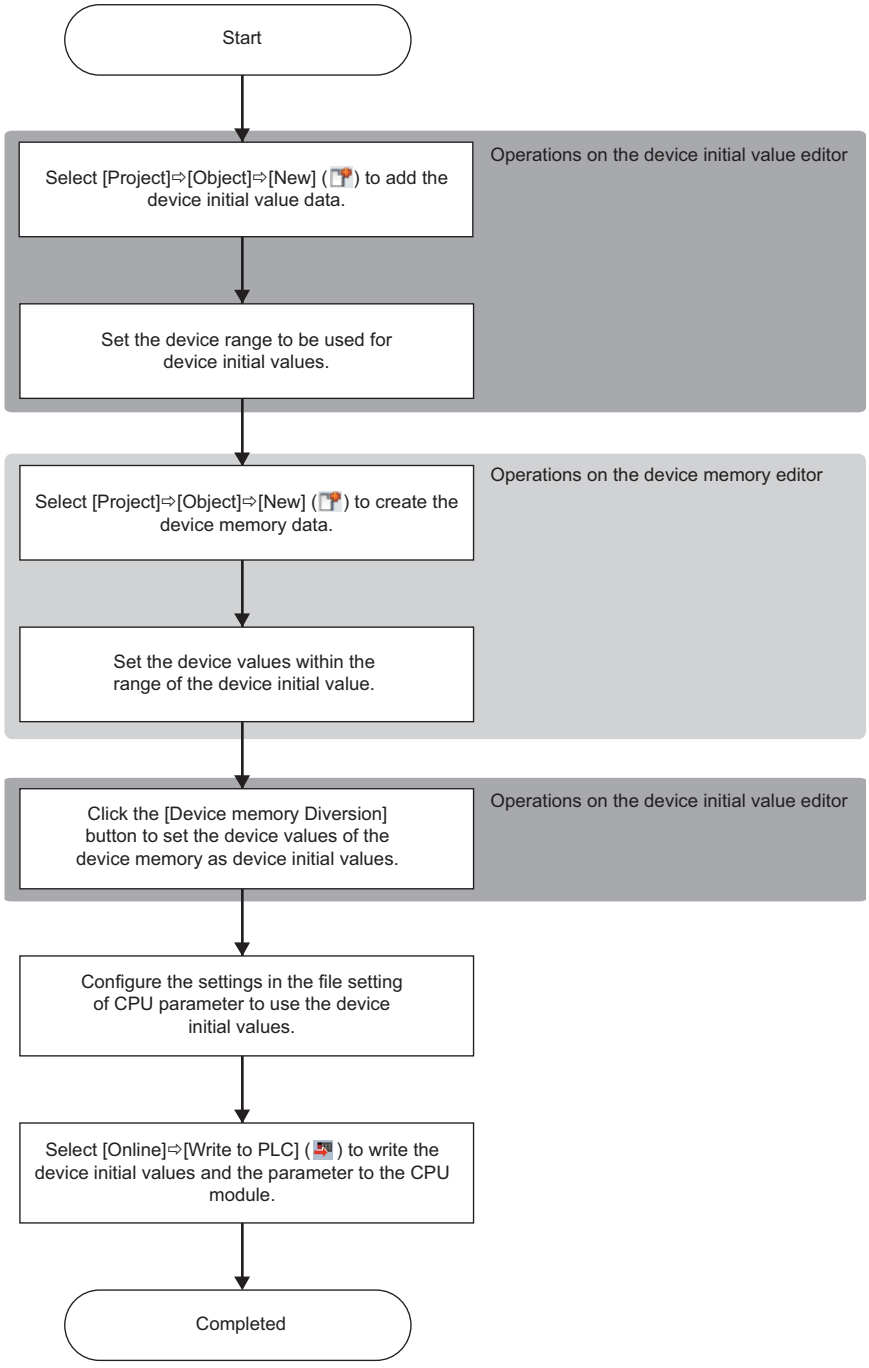
Point

When editing start/last/comment column, press the Shift + F10 keys to display the shortcut menu.

8.3 Setting Device Initial Values

This section explains how to set device values set in the device memory as device initial values.

Setting procedure for device initial values


Start
Operations on the device initial value editor
Select [Project]⇨[Object]⇨[New] ( ) to add the device initial value data.
Set the device range to be used for device initial values.
Operations on the device memory editor
Select [Project]⇨[Object]⇨[New] ( ) to create the device memory data.
Set the device values within the range of the device initial value.
Operations on the device initial value editor
Click the [Device memory Diversion] button to set the device values of the device memory as device initial values.
Configure the settings in the file setting of CPU parameter to use the device initial values.
Select [Online]⇨[Write to PLC] ( ) to write the device initial values and the parameter to the CPU module.
Completed

Setting device initial values

Set the device values set in the device memory as device initial values.
Create a device initial value data in advance. (☞ Page 72 Creating projects)

Operating procedure

1. Set the items on the Device Initial Value editor.
2. Select the data to be utilized from the pull-down list in "Device Memory for Register Diversion", and click the [Register to Device Memory] button.
3. Click the [OK] button.

■Using device initial values

If using the device values of which ranges have been set as initial values at the CPU module startup, specify the file name to be used for the initial values in the "File Setting" tab of the CPU Parameter.

■Device range

The devices that can be set for device initial values are within the range set to "Memory/Device Setting" of the CPU Parameter.

Registering to device memory

The data range set on the Device Initial Value editor is registered to device memory.

Operating procedure

1. Select "Device Memory for Register Diversion".
2. Click the [Register to Device Memory] button.

If the registered data is modified on the device memory, it will not be applied to device initial values. Click the [Device Memory Diversion] button on the Device Initial Value editor again.

Point

To check the values of the device initial values, create a new device memory data and perform "Register to Device Memory" to it.

9 SEARCHING DATA

This section explains how to search for and replace character strings, devices, labels, instructions, etc. in each editor.

Function	Supported editor	Purpose	Reference
Find Device/Label Replace Device/Label	• Ladder editor • ST editor • FBD/LD editor • SFC diagram editor • Label editor	Used to search for/replace devices or labels in a program.	Page 202 Searching/Replacing Devices/Labels
Find Instruction/ Replace Instruction	• Ladder editor • ST editor • FBD/LD editor • SFC diagram editor	Used for searching for/replacing instructions in a program.	Page 204 Searching/Replacing Instructions
Find String/Replace String	• Ladder editor • ST editor • FBD/LD editor • SFC diagram editor • Device comment editor • Label editor	Used for searching for/replacing character strings used in a program, label, and device comment.	Page 205 Searching/Replacing Character Strings
Change Open/Close Contact	• Ladder editor • FBD/LD editor • SFC diagram editor	Used for switching the contact type of a specified device in a program from an open contact to a close contact, or from a close contact to an open contact.	Page 207 Changing Contacts Between Open Contact and Close Contact
Device Batch Replace	• Ladder editor • ST editor • FBD/LD editor	Used for searching for/replacing devices or labels in a program in batch.	Page 208 Batch Replacing of Devices and Labels
Cross Reference	• Ladder editor • ST editor • FBD/LD editor • SFC diagram editor • Label editor • Parameter editor	Used to check the device and label declaration location and referenced location in a list.	Page 210 Creating/displaying cross reference information
Device List		Used to check the usage of devices.	Page 212 Displaying Device Usage List
Search/Replace in a program	• Ladder editor	Used to search for devices and labels, or move to the corresponding row by specifying a Step No. in a ladder program.	Page 143 Searching for/replacing data in program
	• ST editor	Used to move to the corresponding row by specifying a line number on an ST editor.	Page 150 Searching for/replacing data in program
	• SFC diagram editor	Used to search for devices and labels, or move to the corresponding row by specifying a block number or a block name in an SFC block list.	Page 175 Searching for/replacing programs

- The Find/Replace window can also be displayed by selecting [View] ⇨ [Docking Window] ⇨ [Find/Replace] ( ).
- Search/replace can also be switched by selecting from the pull-down list on the "Find and Replace" screen.
- One point of device can be searched/replaced when searching/replacing labels.
- The search starts from the top of the program in the inline structured text box regardless of the cursor position.

Precautions

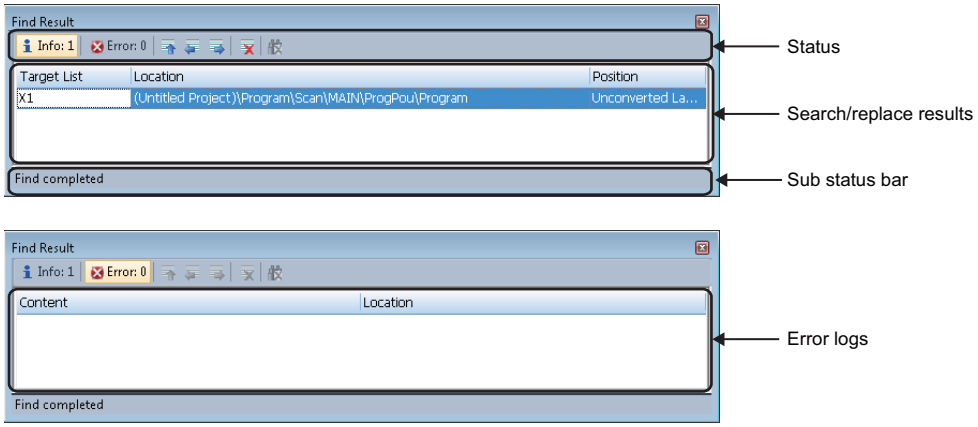
- The program will be in converted state after the replacement.
- The Replace function cannot be performed during monitoring. Perform the function after ending the monitoring. However, it can be performed when the ladder editor is in the Monitor write mode.
- The Find/Replace function cannot be performed for the read-protected data. Perform the function after setting the data in editable mode by disabling the security.

Displaying results and error logs

After the All Find/Replace All function is performed, the results and error logs are displayed.

Window

Click the [All Find] button/[Replace All] button on each Find/Replace screen.



Displayed items

Item	Description
Status	Displays the number of error logs and the search error.
Search/replace results	Displays the position information to specify the location where the search/replace function is performed.
Sub status bar	Displays the search/replace result.
Error logs	Displays the errors in search/replace results.

Point

The jump function is used to jump from any row of search/replace results or logs to the corresponding character strings.
Right-click the selected row and select [Jump] from the shortcut menu, or double-click the selected line.

9.1 Searching/Replacing Devices/Labels

Search for/replace devices/labels in the program.

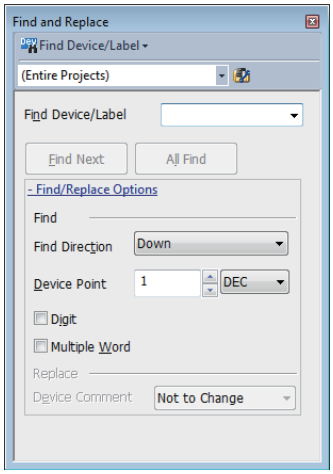
Supported editors: Ladder editor, ST editor, FBD/LD editor, SFC diagram editor, and label editor

When searching a device name, the global label to which the device is assigned will also be found.

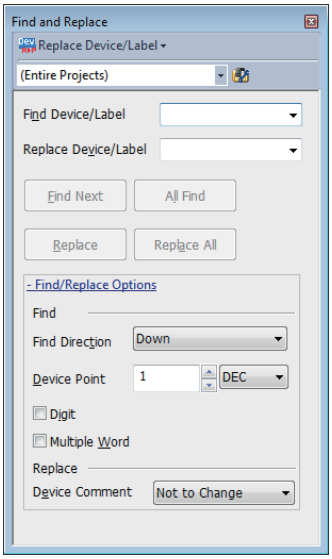
Window

[Find/Replace] ⇨ [Find Device/Label]/[Replace Device/Label] ( )

Find Device



Replace Device



Operating procedure

Set the items on the screen and click the [Find Next] button.

Item			Description
Find/Replace Options	Search	Device point	Enter the number of points to be searched/replaced, counted from the device entered in the "Find Device/Label" field. When a label is entered for "Find Device/Label" or "Replace Device/Label", enter 1. Example) When X50 for "Find Device", X100 for "Replace Device", 3 for "Device Point", and "DEC" for entered value are set The devices are replaced as follows: X50→X100, X51→X101, X52→X102
		Digit	Select this to search for the entered device and digit-specified bit devices that include the entered device.
		Multiple word	Select this to search for the entered device and the double-word format word devices that include the entered device.
	Replace	Device comment	Select this whether to move a device comment of "Find Device/Label" to "Replace Device/Label" (excluding S/TR). If multiple comments have been set, comments of a display target only can be edited.

Examples of device search

■Options

• Option: None

Device specification	Search result
M0	M0, K4M0, M0Z0, K4M0Z0
K4M0	K4M0, K4M0Z0
D0	D0, D0Z0, D0.1
D0.1	D0.1
J1\B0	J1\B0, J1\B0Z0, J1Z0\B0, J1Z0\B0Z0, J1\K4B0, J1\K4B0Z0, J1Z0\K4B0, J1Z0\K4B0Z0

• Option: Digit

Device specification	Search result
X0 to X3	K1X0
X0 to X0F	K4X0
X0 to X1F	K8X0
X0Z0	X0Z0, K1X0Z0, K4X0Z0, K8X0Z0

• Option: Multiple words

Device specification	Search result
D0 to D1	DMOV K1 D0, EMOV E1 D0, MOV K1 @D0
D0 to D9	BMOV D0 D100 K10
J1\W0 to J1\W1	DMOV K1 J1\W0
@D0 to @D1	DMOV K1 @D0
T0 to T1	DMOV K1 T0

■Device suffix display (for ST editor, FBD/LD editor, and SFC diagram editor only)

Device specification	Search result	Remarks
D100	D100: = 1; D100: D: = 1; D100: E:= 0.1;	Devices are searched regardless of suffix.
D100: D	D100: = 1; D100: D: = 1; D100:E := 0.1;	Only specified devices with suffix are searched.

9.2 Searching/Replacing Instructions

Search for/replace instructions in a program.

Supported editors: Ladder editor, ST editor, and FBD/LD editor (LD element and FBD element (FB/FUN) only), and SFC diagram editor

The NOP instructions used in the ladder programs cannot be searched.

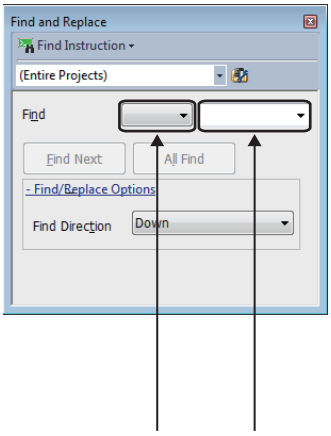
The following application instructions can also be searched as an output instruction.

SET, RST, PLS, PLF, FF, SFT, SFTP, MC

Window

[Find/Replace] ⇨ [Find Instruction] ( )/[Replace Instruction]/[Find Contact or Coil] ( )

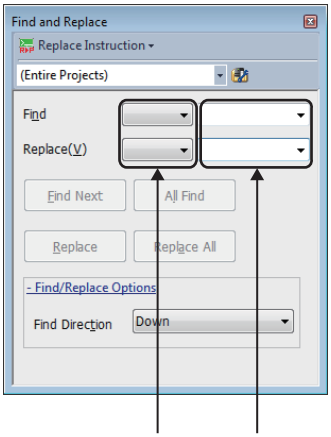
Find Instruction



Element selection area

Device instruction input area

Replace Instruction



Element selection area

Device instruction input area

Operating procedure

Set the items on the screen and click the [Find Next] button.

Examples of instruction search

Example of instruction specification*1	Search result
MOV	MOV, MOVP
MOVP	MOVP
MOV D0 K4Y0	MOV D0 K4Y0, MOVP D0Z1 K4Y0, MOV D0 K4Y0Z1, MOVP D0Z1 K4Y0Z1
MOVP D0 J1\W0	MOVP D0 J1\W0, MOVP D0Z1 J1\W0, MOVP D0 J1Z1\W0Z1, MOVP D0Z1 J1Z1\W0Z1

*1 The search specifying an argument can only be performed in ladder programs.

9.3 Searching/Replacing Character Strings

Search for/replace character strings of the data in the project.

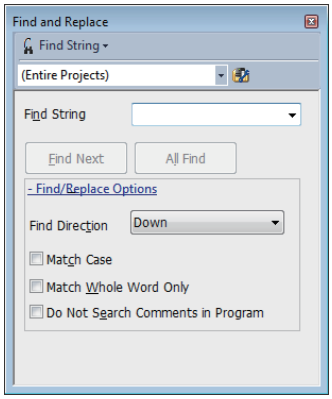
Target editor	Searchable character string	Replaceable character string
Ladder editor	All character strings *1	Character strings except for instruction names
ST editor	Character strings except for the following • Monitoring part (displayed on the right side of the screen during monitoring) • The omitted portion of the collapsed display (example: IF...END_IF)	
FBD/LD editor	All character strings	Character strings except for the following • Data type of a function • Data type of a function block • Input/output label of FB/FUN • Character string, "RETURN" of a return element
SFC diagram editor	All character strings	Character strings except for the following • Qualifier • Comment of Step No./Transition No. • Step name specified to the jump destination
Label editor	Character strings except for class name	Character strings except for the following • Class name, data type, device name
Device comment editor	Character strings in the comment column	

*1 The NOP instructions used in the ladder programs cannot be searched.

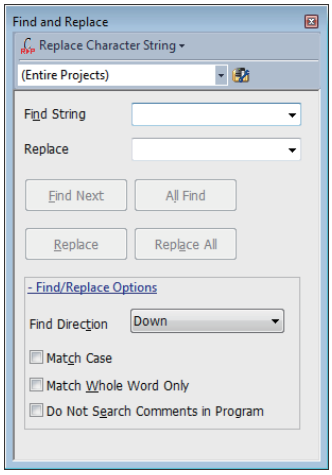
Window

[Find/Replace] ⇨ [Find String]/[Replace String]

Find String



Replace String



Operating procedure

Set the items on the screen and click the [Find Next] button.

Search/replace data by matching whole word only

The whole word means a string delimited by the break characters.

The character string which is completely match with the character string entered in "Find String" is searched.

Break characters are shown as below.

- Space, tab, line feed, operator in ST editor

Ex.

Search for a device comment 'abc ; def' by the following character strings.

String to be searched for	Search result	
	Unselected	Selected
a	abc; def	Nothing is found.
abc	abc; def	abc; def
bc	abc; def	Nothing is found.
abc ; def	abc; def	abc; def

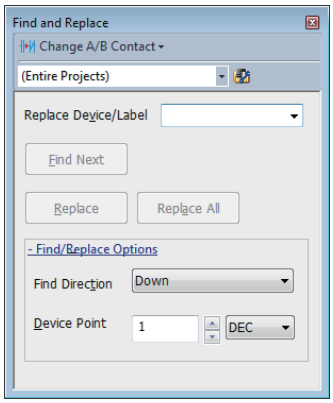
9.4 Changing Contacts Between Open Contact and Close Contact

Change contact types from open contact to close contact, and vise versa.

Supported editors: Ladder editor, FBD/LD editor, and SFC diagram editor

Window

[Find/Replace] ⇨ [Change Open/Close Contact]



Operating procedure

1. Set the items on the screen and click the [Find Next] button.

Item		Description
Find/Replace Options	Device Point	Enter the number of points to be replaced, counted from the device/label entered in the "Replace Device/Label" field. When a label is entered for "Replace Device/Label", enter 1. Example) When X100 for "Replace Device", 3 for "Device Point", and "DEC" for entered value are set The open/close contact of X100, X101, and X102 will be replaced.

2. Click the [Replace] or [Replace All] button to change the contact type.

9.5 Batch Replacing of Devices and Labels

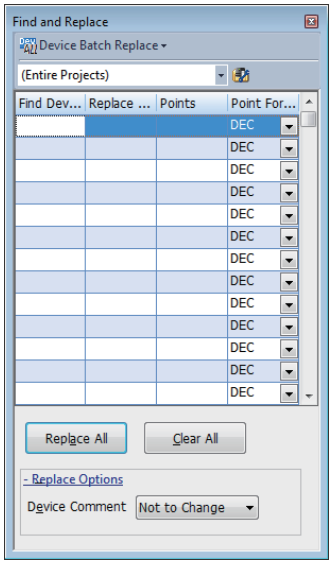
For devices that can be replaced in batch, refer to the following section.

☞ Page 202 Searching/Replacing Devices/Labels

Supported editors: Ladder editor, ST editor, FBD/LD editor, and SFC diagram editor

Window

[Find/Replace] ⇨ [Device Batch Replace]



Operating procedure

Set the items on the screen, and click the [Replace All] button.

Item		Description
Points		Points: Enter the number of points to be replaced, counted from the device specified for "Find Device". When a label is entered for "Find Device" or "Replace Device", enter 1. Example) When X0 for "Find Device", X10 for "Replace Device", 5 for "Points", and "DEC" for "Point Format" are set The devices are replaced as: X0 → X10, X1 → X11, X2 → X12, X3 → X13, X4 → X14
Point Format		
Replace Options	Device Comment	Select whether to copy/move a device comment in "Find Device" to "Replace Device" (excluding S/TR).

Point

Select a range of device/label on the program editor by a drag-and-drop operation to register multiple devices/labels in batch.

Precautions

When specifying S□/TR□ for "Find Device" and select "(Entire Projects)" for the target, S□/TR□ of all blocks are replaced.
(Example: when replacing 'TR0', BL0\TR0 and BL1\TR0 also are replaced.)

9.6 Displaying Reference Information List of Devices/Labels

Search for devices/labels and display information where the selected devices/label are used in the project in a list.
The Cross Reference window is displayed horizontally when it is docked on top or bottom of the main frame, and displayed vertically when it is docked on left or right of the main frame.

Point

The display format and detailed operations for each function can be set by the following option setting.
[Tool] ⇨ [Options] ⇨ "Find/Replace" ⇨ "Cross Reference"

Target data for creating cross reference information

Cross reference information is created based on devices used in the following data.

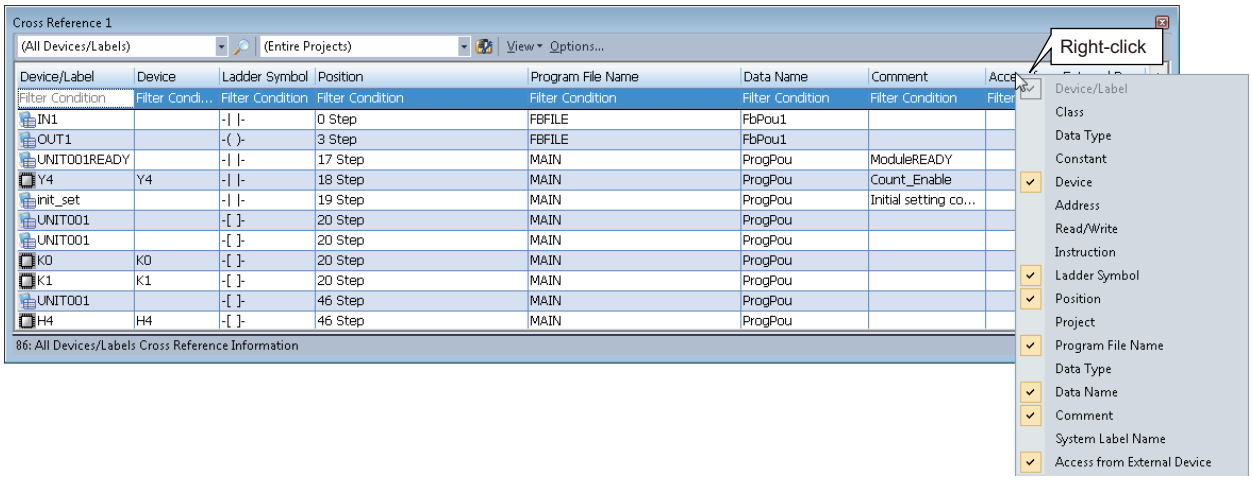
- Ladder
- ST
- FBD/LD
- SFC
- Global label
- Local label
- Structure
- Multiple CPU refresh setting of the CPU parameter
- Refresh setting of module parameters

Creating/displaying cross reference information

Create and display the cross reference information of the devices/labels used in the project.
Supported editors: Ladder editor, ST editor, FBD/LD editor, SFC diagram editor, label editor, and parameter editor

Window

[View] ⇨ [Docking Window] ⇨ [Cross Reference1/Cross Reference2] ( )



Up to 2 Find result screens can be displayed at once.
Up to 80,000 creation results of the cross reference information is displayed.

Operating procedure

1. Set the display format from the pull-down list displayed by clicking [View] as necessary.

2. Enter a device/label to be searched for in (All Devices/Labels) column.

To specify the range to be searched, select the search location of devices/labels by clicking on the tool bar. (Multiple selections allowed.)

Press the Enter key or double-click or right-click with selecting a row on the result list, then select [Jump] from the shortcut menu. The editor, in which the devices/labels are used, opens and the corresponding devices/labels are selected.

Point

- Pressing the F12 key (Ctrl + .) moves the focus between the editor and the Cross Reference window.
- Pressing the F11 key (Ctrl + .)/Shift + F11 keys (Ctrl + Shift + .) moves the cursor to the previous/next row.

■Displaying comments

For a ladder, the comments specified in [Tool] ⇨ [Options] ⇨ "Project" ⇨ "Device Comment Reference/Reflection Target" are displayed as device comments.

☞ Page 176 Device comments

■Creating cross reference information

Cross reference information can also be created by right-clicking a device/label on the program editor, and selecting [Cross Reference] from the shortcut menu.

After the Cross Reference window is displayed, cross reference information can be created based on all devices/labels by selecting "(All Devices/Labels)" and press the Enter key.

■Automatic synchronization

A function to create the cross reference information of devices/labels selected on the editor automatically.

The Cross Reference window which displays the result of automatic synchronization can be specified in "Operational Setting" of "Option" by selecting [Option] on the Cross Reference window.

Displaying cross reference information

■Display after creation of cross reference information is completed

- The data protected by security function cannot be displayed.
- If the program or the option setting is changed, the cross reference information does not correspond with the program. In order to update the cross reference information, create it again.
- The columns can be sorted in ascending/descending order by clicking the column header. However, the sort cannot be performed if the structure/array/function block is displayed in the tree.
  To clear the display in the tree, unselect [Display Hierarchically] from the pull-down list of [View].

■When data is deleted after conversion

The deleted data may be displayed as a search result.

To display the most recent cross reference information, convert the program and create the cross reference information again.

Filtering display

Filter the created cross reference information.

Operating procedure

Enter/select the filter condition, and press the Enter key.

■Filtering condition

The previously entered keywords can be selected from the pull-down list.

For "Data Name" and "Program File Name", search results can also be selected from the list as well as previously entered keywords.

■Tree display of filtering condition

Only data of which components match with the filtering conditions are displayed in the tree format when filtering display is performed for the structure data names or the instance names of the function block.

■Deleting filtering condition

Filtering display of the column can be canceled by deleting the keyword entered in each filtering condition column.

■Keywords for filtering condition

The wild card characters can be used in a filtering condition as shown below.

Ex.

When the filtering condition is set for the device/label column.

Wild card	Search target	Example	Search result
*	Specify any character string.	*30*	ready301, K4X30, K1Y30, K4Y30
?	Specify any one character.	K4?30	K4X30, K4Y30
[ ]	Specify any one of those characters.	[XY]8	X8, Y8
[!]	Specify any one of characters except for the characters in the bracket.	K4X[!3]0	K4X40
[-]	Specify character strings within the range in the bracket.	D[0-2]	D0, D1, D2

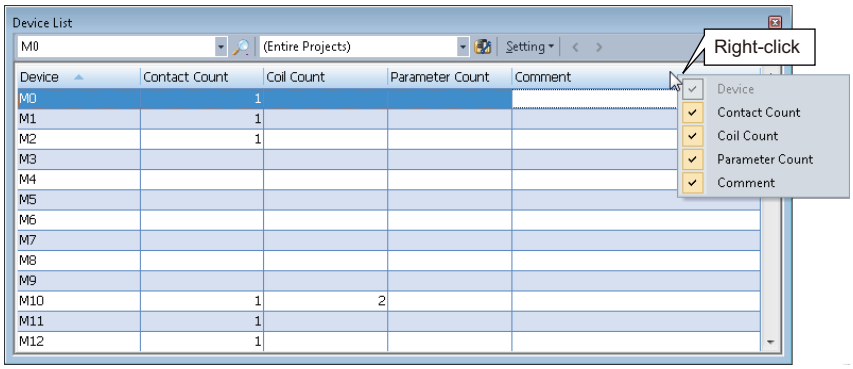
9.7 Displaying Device Usage List

Display the usage of the specified devices.

Supported editors: Ladder editor, ST editor, FBD/LD editor, SFC diagram editor, label editor, and parameter editor

Window

[Find/Replace] ⇨ [Device List].



Operating procedure

Set the items and press the Enter key.

To specify the range to be searched, select the search location of devices/labels by clicking on the tool bar. (Multiple selections allowed.)

Considerations for search in Device List

■Searching for coil instructions

As output instructions, the following application instructions can also be the target of search.

SET, RST, PLS, PLF, FF, SFT, SFTP, and MC

■Searching for R device and ZR device

R device and ZR device are searched separately in the Device List.

Specify them separately when searching for R device or ZR device.

FX5CPU does not support ZR devices.

■Confirming locations of devices

The location of the device used can be checked by opening the Cross Reference window from the device in the Device List.

To open the Cross Reference window, select a row in the Device List and follow any of the following operations.

- Press the Ctrl+E keys or press the Enter key
- Double-click the row
- Right-click ⇨ select [Cross Reference] from the shortcut menu

For details on the Cross reference function, refer to the following section.

☞ Page 209 Displaying Reference Information List of Devices/Labels

■Displaying comments

- When searching for data with selecting one program, the comments specified by selecting [Tool] ⇨ [Options] ⇨ "Project" ⇨ "Device Comment Reference/Reflection Target" are displayed. (☞ Page 176 Device comments)
- When searching for data with selecting multiple programs, common device comments are displayed.

■Searching for a step relay (S☐)

when searching for a step relay (S☐), specify a step relay with a block specification (BL☐\S☐).

10 SEGMENTING PROGRAMS

Segmenting a processing program which is used repeatedly in programs into elements, the elements can be utilized in sequence programs.
The efficient program development reduces program errors, thus the program quality will be improved.
Segmented program file is referred to as POU (☞ Page 58 Data configuration).
Besides, multiple POUs can be collected into a separate file from a project. The file is referred to as a library.

POUs

Two types of POU are able to utilize repeatedly; function blocks and functions.

■Function block

Function block is a program component (POU) created by segmenting a program, which outputs an operation result in accordance with a value of the internal memory and an input value, into components.
It is classified into the following three types.

POU	Description	Reference
Function block (User-created)	A function block which is created from a program using the labels which are defined as input labels, internal labels, and output labels.	Page 214 Creating Function Blocks
Standard function block	A function block which can be used in general. It is prepared in GX Works3 in advance.	Page 32 Element Selection window
Module FB	A function block which is created by segmenting a dedicated processing for modules. It is prepared in GX Works3 in advance.	Page 216 Enhanced use of module FBs

■Function

Function is a program component (POU) created by segmenting a program, which always outputs the same operation results against the same input value without internal memory.
It is classified into the following two types.

POU	Description	Reference
Function (User-created)	A function block which is created from a program using the labels which are defined as input labels and output labels	Page 217 Creating Functions
Standard function	A function which can be used in general. It is prepared in GX Works3 in advance.	Page 32 Element Selection window

Library

Library is an element collection of multiple POUs and structures. The elements in the library can be used in multiple projects.
FX5CPU does not support it.

■User library

User library is an element collection which is composed of created POUs and structures.
For details, refer to the following section.
☞ Page 219 Enhancing Use of User Library

■Application library/sample library (MELSOFT Library)

MELSOFT Library (Sample Library) is an element collection that has been prepared by maker in advance.
To obtain the application library/sample library (MELSOFT Library), please consult your local Mitsubishi representative.
For details, refer to the following section.
☞ Page 223 Enhanced Use of Application Library/Sample Library

10.1 Creating Function Blocks

This section explains how to create sequence programs using function blocks.

Function blocks

For details of the function blocks, refer to the following manuals.

📖 MELSEC iQ-R Programming Manual (Program Design)

📖 MELSEC iQ-F FX5 Programming Manual (Program Design)

Procedure to use

1. Create a data of function block. (☞ Page 214 Creating function block data)
2. Set the local labels to be used in the program of a function block. (☞ Page 215 Setting labels)
3. Create a program using labels. (☞ Page 215 Creating function block programs)
4. Paste the function block on the sequence program. (☞ Page 128 Inserting function blocks, Page 148 Inserting function blocks)

Creating function blocks

The following explains how to create function blocks.

■ Number of function blocks that can be created

RCPU (other than R04CPU): Up to 8192 in conjunction with function data

R04CPU: Up to 4096 in conjunction with function data

FX5CPU: Up to 960 in conjunction with function data

However, the maximum number of data that can be created may not be reached depending on the configuration of FB files and FUN files.

Precautions

When changing line statements of a function block on the tree, all programs in which the changed FB instances are used will be in the unconverted state.

Creating function block data

Create a function block data in the project on the "New Data" screen.

Item				Description
Detail Setting	Inherent Property	Use MC/MCR in EN Control*1	Yes	Applies function block program codes using the MC/MCR instruction to each location where instances are used at conversion.
			No	Applies function block program codes using the CJ instruction to each location where instances are used at conversion.
		Use EN/ENO	Yes	Becomes a function block with EN/ENO. EN/ENO labels can be used in projects without registering in the local labels. For details on EN/ENO, refer to the following manuals. 📖 MELSEC iQ-R Programming Manual (Program Design) 📖 MELSEC iQ-F FX5 Programming Manual (Program Design)
			No	Becomes a function block without EN/ENO.
	FB File	FB Type	Macro Type	Stores the program of a function block to the FB file to which the call source program block or program of the function block is stored.
			Subroutine Type	Stores the program of a function block to an FB file.
		FB File of Add Destination		Select a storage destination file of a function block to be created. The storage destination files can be created by entering file name directly.

*1 This item can be selected only when both "Yes" is selected to "Use EN/ENO" and "FB Type" is "Macro Type".

Precautions

If the function block in which the MC/MCR instruction is used is not executed, the outputs and the current values of timers in the function block will be reset. (If the instruction is not used, the current values are retained).

Setting labels

Set the labels to be used in the program on the label editor.

The operations on the screen is the same as that of other label editor. For details, refer to the following section.

☞ Page 107 Registering Labels

Operating procedure

Select "FB/FUN" ⇨ "(file name)" ⇨ "(FB)" ⇨ "Local Label" on the Project view.

Creating function block programs

Create a program of a function block using labels.

Both local labels of the function block and global labels can be used.

The method for entering programs is same as that of each program.

Operating procedure

1. Select "FB/FUN" ⇨ "(file name)" ⇨ "(FB)" ⇨ "Program" on the Project view.
2. Configure the programs.

Converting function blocks

- Function blocks (POUs) will not be in the converted state even if the conversion was attempted as long as FB instance has not been created, or FB instance has been created only in unregistered programs.
- The FB file will be in the unconverted state when function blocks are edited. Once the FB file is changed from the converted state to the unconverted state, all programs in which the function blocks are sued will be in the unconverted state.

Considerations when creating FB programs

■Use of devices

Using labels is recommended for creating FB programs.

If an FB program which uses devices (X10, Y10, etc.) is used at multiple locations, the program may not operate normally.

In addition, if an FB program which uses devices for the OUT instruction is used at multiple locations, duplicated coil is resulted. Duplicated coil can be avoided with the SET/RST instruction.

■When using master control instructions

When using a master control instruction in an FB program, use the MC instruction and the MCR instruction together.

■When using function blocks for a sequence program to be executed multiple times in a single scan

If a function block, which contains a rising instruction/falling instruction, or an instruction that requires multiple scans before the execution completion, is used in a sequence program to be executed multiple times in a single scan, the program may not operate normally.

Item	Description
Program type to be executed multiple times in a single scan	• Fixed scan execution type program • Interrupt program • Subroutine program • FOR-NEXT instruction program
Unusable instructions for FBs which are used for above programs	Instructions which require multiple scans before the execution completion (JP.READ/JP.WRITE instruction, SORT instruction, SP.FREAD/SP.FWRITE instruction, etc.)
	Rising instruction (□P instruction (such as MOVP instruction), PLS, etc.)
	Falling instruction (PLF, LDF, ANDF, ORF, MEF, FCALLP, EFCALLP, etc.)

Enhanced use of module FBs

In GX Works3, Module FB which is composed of the segmented processing for each module (function block), is prepared. By using the module FBs, the operation settings can easily be performed without programming the processing of each module.

Note that the module FBs cannot be used in safety programs.

Importing module FBs in project

Module FBs are displayed in the [Module] tab in the Element Selection window when one of the following operations is performed.

- Place the modules on the Module Configuration Diagram and fix the parameters
- Add a new module from the Project view

Point

The manual of a module FB can be displayed from the Element Selection window.
Select and right-click a module FB, then select [Help] from the shortcut menu.

Using module FBs in programs

Operating procedure

1. Drag and drop a module FB from the [Module] tab of the Element Selection window onto the program.
2. Enter an FB instance name.

After creating the FB instance, the created function block (Local Label, Program) is registered in "M_FBLIB" on the Project view.

Precautions

Registering the module FBs to be used in advance (at creation of new project) is recommended.
If the module FB to be used is not registered, it may take time to display the module FB after dragging and dropping it.

Setting operation parameter of module FB

After creating FB instance of a module label, display it on the label editor and enter the value in "Initial Value" displayed in "Extended Display".

However, the initial value of an array type label cannot be set in the operation parameter. Set the operation parameter by a program.

When the operation parameter of module FB set in the label initial value is changed by program, the module FB operates with the changed value. Before changing the label values of a module FB, check the influence of the change using the Cross Reference function.

Editing module FB

The local label and program of a module FB cannot be edited.

However, a module FB can be copied in the Project view and it can be edited.

The name of copied module FB is changed from '+' to '_'.

10.2 Creating Functions

This section explains how to create a sequence program using functions.

Functions

For details of the functions, refer to the following manuals.

📖 MELSEC iQ-R Programming Manual (Program Design)

📖 MELSEC iQ-F FX5 Programming Manual (Program Design)

Procedure to use

1. Create a data of function. (☞ Page 217 Creating function data)
2. Set the local labels to be used in the FUN program. (☞ Page 217 Setting labels)
3. Create a FUN program using labels. (☞ Page 218 Creating function programs)
4. Paste the function onto a sequence program. (☞ Page 133 Inserting functions, Page 148 Inserting functions)

Creating functions

The following explains how to create functions.

■ Number of function blocks that can be created

RCPU (other than R04CPU): Up to 8192 in conjunction with function block data

R04CPU: Up to 4096 in conjunction with function block data

FX5CPU: Up to 960 in conjunction with function block data

However, the maximum number of data that can be created may not be reached depending on the configuration of FB files and FUN files.

Creating function data

Create a function data in the project on the "New Data" screen.

Item			Description
Detail Setting	Use EN/ENO	Yes	Become a function with EN/ENO. EN/ENO labels can be used in the project without registering in the local labels. For details on EN/ENO, refer to the following manuals. 📖 MELSEC iQ-R Programming Manual (Program Design) 📖 MELSEC iQ-F FX5 Programming Manual (Program Design)
		No	Become a function with EN/ENO.
	FUN File of Add Destination		Select a storage destination file of a function to be created. The storage destination files can be created by entering file name directly.

Setting labels

Set the labels to be used in the program on the label editor.

The operations on the screen is the same as that of other label editor. For details, refer to the following section.

☞ Page 107 Registering Labels

Operating procedure

Select "FB/FUN" ⇨ "(file name)" ⇨ "(Function)" ⇨ "Local Label" on the Project view.

Creating function programs

Create a program of a function using labels.

The labels set to the local label of the function can be used.

The method for entering programs is same as that of each program.

Operating procedure

1. Select Project view ⇨ "FB/FUN" ⇨ "(file name)" ⇨ "(function)" ⇨ "Program".
2. Configure the programs.

Converting functions

- Functions (POUs) will not be in the converted state even if the conversion was attempted as long as functions are not used in the program, or functions are used only in unregistered programs.
- The FUN file will be in the unconverted state when functions are edited. Once the FUN file is changed from the converted state to the unconverted state, all programs in which the functions are used will be in the unconverted state.

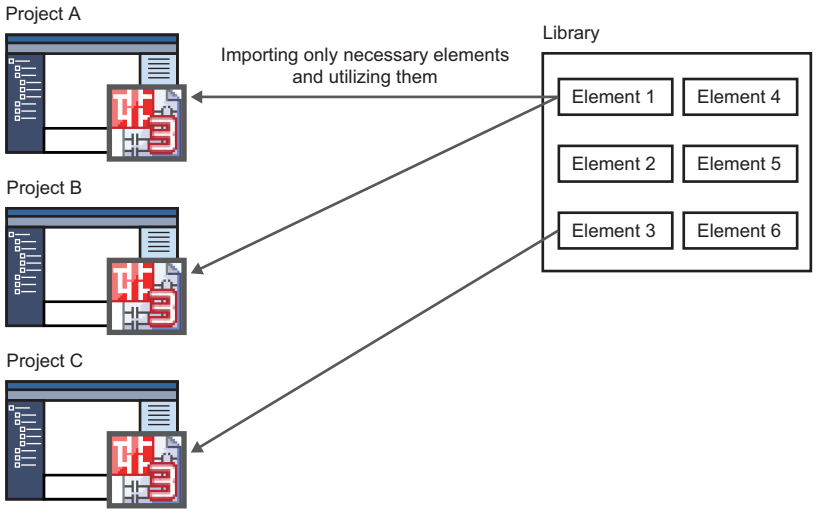
10.3 Enhancing Use of User Library

This section explains how to store created elements in a library file and utilize them in sequence programs.
In this section, the user library is explained as a library.

User library

User library collects the elements which are used generally in multiple projects to make them easy to use. It can be used by importing elements in the library to each project.
The following data can be registered in the user library.

- POUs (Program block, Function block, Function)
- Global labels, structures



Procedure to use

1. Create a library file. (☞ Page 220 Creating libraries)
2. Register a library in GX Works3. (☞ Page 221 Registering to library list)
3. Utilize elements from a library to projects. (☞ Page 222 Utilizing elements)

Precautions

When using the library file which is on a network drive or a removable media, save the file in the hard disk of a personal computer before using.

Creating libraries

The following explains about the operations of library such as how to creating and editing.

Preparing a source project for library

To create a library, a source project is required.
Prepare an existing project or a project that collects the elements to be registered in a library.

Precautions

After registering a library, the registered elements are displayed in the [Library] tab on the Element Selection window.
However, the contents of global labels (such as label name, data type) are not displayed.
When creating a library that includes multiple global labels, set comments to distinguish them on the "Properties" screen of global labels.

Creating library file

A library file can be created by exporting project as a library file (*.usl).
Library file is a file in which data related with elements in projects is stored.
To prevent libraries and elements from manipulation, set the password when exporting them.

Operating procedure

1. After creating elements in the project, select [Project] ⇨ [Library Operation] ⇨ [Export Library].
2. Set each item on the confirmation screen displayed, then click the [OK] button.

The operating availability of elements which is used in a project of utilization destination, can be set by setting a password to edit on the above screen.
If it is not set, the contents of utilized elements can be accessed.
○: Available, ×: Not available

Item		Description	Operation for element		
			Use	Display	Edit
Enable to read. • Disable editing POU. (Change to read-only attribute.)	Unselected	It is used to edit the contents of programs or labels when ,for example, distributing samples or guaranteeing the operations by users.	○	○	○
	Selected	To prevent from manipulation, the contents of programs and labels are opened, but not editable.	○	○	×
Disable to read.		To prevent from leakage, the programs are not opened. The contents of labels (including structures) are opened, but not editable.	○	×	×

Precautions

The utilized elements from other library maintain the settings when it was utilized regardless of the reading availability set when they were exported. (Example: When the read-protected element A is utilized and elements including the element A are exported with the settings that enables to read. That time, the element A will be read-protected.)

■Creating a library in GX Works3 format from a library in GX Works2 format

Operating procedure

1. Select [Project] ⇨ [Open Other Format File] ⇨ [GX Works2 Format] ⇨ [Open User Library], and specify the library to be converted into a GX Works3 project.
2. Select the series on the confirmation screen displayed, then click the [OK] button.
3. Check the project after changing the module type, then edit it if necessary.
4. Export the project as a library file.

Editing library

Elements in a library file can be edited, added and deleted same as projects.

The password authentication is required when editing the library protected by password. Besides, the user registration is required for the library file exported from a safety project. (☞ Page 305 User management)

For details on the maximum number that can be created for each data, refer to the following section.

☞ Page 71 Maximum number of data that can be created in one project

Operating procedure

1. Select [Project] ⇨ [Open Other Format File] ⇨ [GX Works3 Format] ⇨ [Open User Library].
2. Edit POUs.

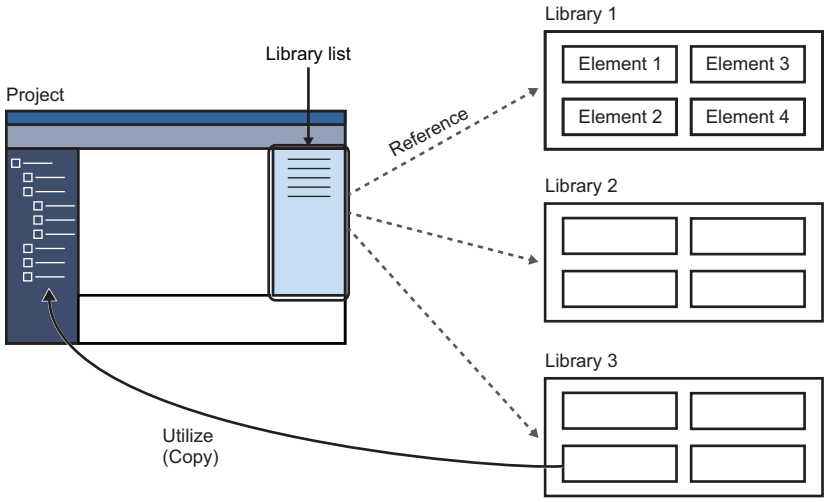
The editing method is same way as elements of the project.

Enhanced use of libraries

Element in a library can be utilized by registering the library into the library list.

The reference to the library file is registered in the library list.

The information of library file is saved not in a project unit but in a logon user unit on personal computer.



For the considerations for utilizing the library file created in the different version of GX Works3, refer to the following section.

☞ Page 340 Use Project in Different Version

Registering to library list

Register the library, which contains the elements to be utilized, to the list.

Up to 64 libraries can be registered.

When the registered library file was edited while GX Works3 is running , updating the display information of library is necessary.

Operating procedure

■Registering library to be referenced

1. Select the [Library] tab on the Element Selection window.
2. Select "User Library", then select [Project] ⇨ [Library Operation] ⇨ [Register to Library List] ( ).
3. Select a file on the "Register Library to Library List" screen, then click the [Open] button.

For the library/element which is protected by password, the icon will be displayed in a pale color on the [Library] tab.

■Deleting library/updating display information

1. Select the [Library] tab on the Element Selection window.
2. Select the library to be deleted/updated, then select [Project] ⇨ [Library Operation] ⇨ [Delete from Library List] ( )/ [Update the Display Information of Library].

Utilizing elements

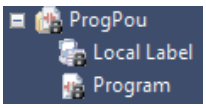
The element registered in the library on the Element Selection window can be utilized in other projects.
To utilize POU that was set the read-protection at the time of library export, entering password is required before editing.

Operating procedure

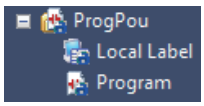
■Pasting elements onto the Project view

Drag and drop an element onto the project view from the [Library] tab on the Element Selection window.
The elements can be dropped only to the place which is able to create the same kind of data.
The read-protected POU is displayed on the Project view as follow.

Read-protected



Authenticated



■Pasting elements onto the program editor (function block/function only)

1. Drag and drop an element onto the sequence program from the [Library] tab on the Element Selection window.
2. When pasting function block, select either the global label or the local label to register the element from the list on the "FB Instance Name" screen.

■Making the read-protected POUs editable

1. Select the POU which was utilized and read-protected, then right-click and select [Enter the Password to Edit] from the shortcut menu.
2. Enter the password on the "Password Authentication" screen, and click the [OK] button.

The element will be editable until the project is closed.

■Copy of utilized elements

- When the data name of the utilized element has "+", it will be changed to "_" after copy.
- The read-protected element cannot be copied. However, it can be copied by entering password to edit.
- For the edit-protected element, the copy of elements only can be editable.

10.4 Enhanced Use of Application Library/Sample Library

Application library/sample library (MELSOFT Library) are the element collection of program/function block/function/structure which are composed of specific processing.
In this section, the application library/sample library (MELSOFT Library) are explained as a library.
The elements in a library can be used by registering the library into the library list.
The elements are imported into a project with by registering a library.
To obtain the library, please consult your local Mitsubishi representative.
FX5CPU does not support it.

Precautions

Before registering a library, confirm that the module type of the current project (CPU module) supports the library.

Registering libraries

The following shows how to register a library.
The methods differ depending on the file extension (*.gx3s, *.mslm).

■When the extension is "gx3s"

Operating procedure

1. Select [Tool] ⇨ [Sample Library Registration].
2. Select a file on the "Open Sample Library" screen, and click the [OK] button.

The function block of the library is displayed on the Project view ('M_FBLIB' is for the Mitsubishi Electric FA products, and 'P_FBLIB' is for the partner's products).

■When the extension is "mslm"

Operating procedure

■Registering library to be referenced

1. Select the [Library] tab on the Element Selection window.
2. Select "Application Library/Sample Library", then select [Project] ⇨ [Library Operation] ⇨ [Register to Library List] ( ).
3. Select a file on the "Register Library to Library List" screen, then click the [Open] button.

The registered library is displayed on the library list.

■Deleting libraries

1. Select the [Library] tab on the Element Selection window.
2. Select the library to be deleted, then select [Project] ⇨ [Library Operation] ⇨ [Delete from Library List] ( ).

■Updating libraries

Delete a library once, then re-register an new library.

MEMO

PART 4 DEBUGGING AND OPERATION

This part explains the communication route setting to access a CPU module, reading/writing data, and monitoring the execution status.

11 SIMULATING PROGRAM

12 SETTING ROUTE TO CPU MODULE

13 WRITING/READING DATA TO CPU MODULE

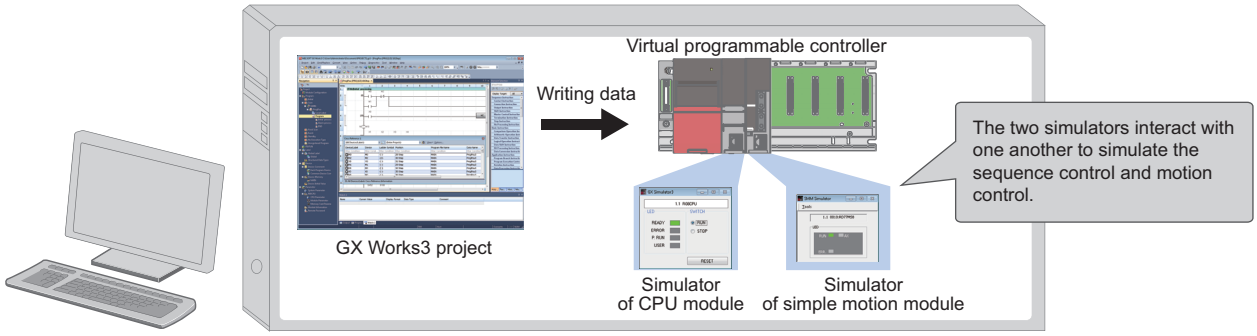
14 CHECKING OPERATION OF PROGRAMS

11 SIMULATING PROGRAM

This chapter explains how to debug programs offline using the Simulation function.
For information on CPU modules supporting the Simulation function, refer to the following section.
☞ Page 342 Using Simulation Function
Note that SFC program is not supported to the Simulation function.

11.1 Simulation Function

The Simulation function debugs programs using a virtual programmable controller on a personal computer.
GX Simulator3 is used for the Simulation function.
This function is useful to check programs before operating them actually since it enables debugging without the connection with CPU module.
For the considerations for using the Simulation function, refer to the following section.
☞ Page 355 Considerations


Virtual programmable controller
Writing data
GX Works3 project
Simulator
of CPU module
Simulator
of simple motion module
The two simulators interact with one another to simulate the sequence control and motion control.

Safety and handling consideration

⚠CAUTION

■The Simulation function simulates the actual module to debug a created sequence program. However, this function does not guarantee the operation of the debugged program.
■The Simulation function uses the memory for simulation to input and output data to/from the I/O module and Intelligent function module. Some instructions/functions and device memory are not supported. Therefore, the operation results obtained with the virtual programmable controller may differ from those obtained using the actual module.

After debugging programs using the Simulation function, the normal program debugging, which is performed with the module connected before the actual operation, is required.

11.2 Starting/Stopping Simulation

Start/stop the Simulation function.
The connection destination is GX Simulator3 during simulation, and "Simulation ((system number of the simulator).(CPU number))" is displayed on the status bar.

Window

[Debug] ⇨ [Simulation] ⇨ [Start Simulation] ( )/[Stop Simulation] ( )

Operating procedure

Select the checkbox of the data to be written on the "Online Data Operation" screen displayed, and click the [Execute] button.

Cooperative simulation

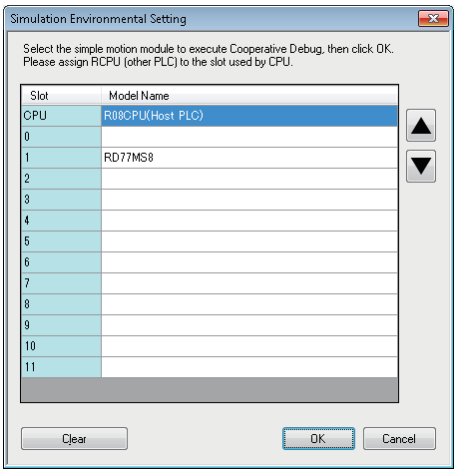
The cooperative simulation between the simulator of a simple motion module (SMM Simulator) and the simulator of GX Works3 (GX Simulator3) is available.

Environmental setting

Specify the target before starting simulation in order to simulate the cooperative operations of a sequence program and a motion control (simple motion module).

Window

[Debug] ⇨ [Simulation] ⇨ [Simulation Environmental Setting] ( )



Operating procedure

Select the target module to be run together, and click the [OK] button.

■In multiple CPU configuration

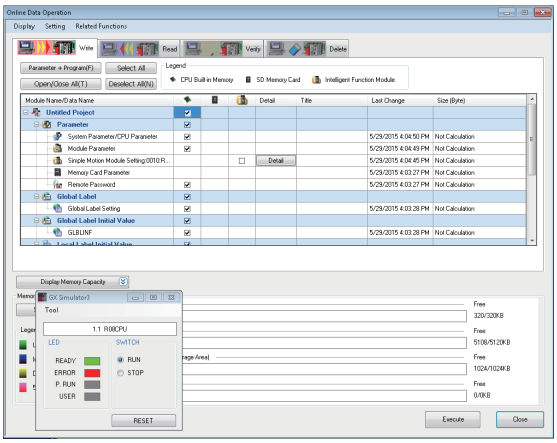
Select the other CPUs in accordance with the module configuration set to the project.

Starting cooperative simulation

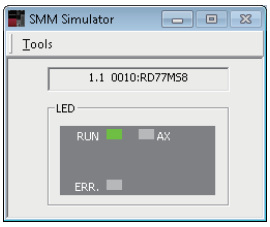
The following shows how to operate GX Simulator3 and SMM Simulator with them interacted with each other.

■Starting simulation

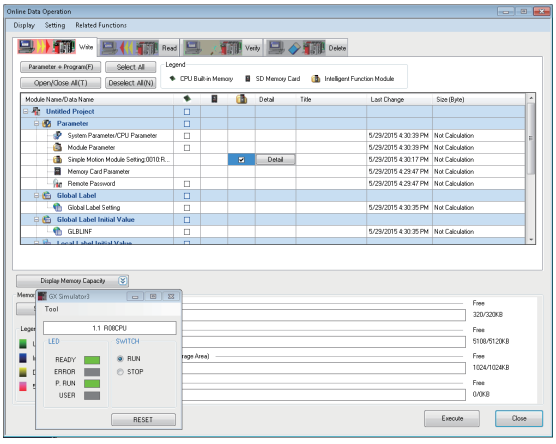
Operating procedure



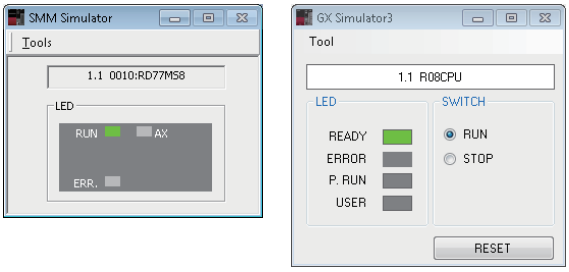
1. Select [Debug] ⇨ [Simulation] ⇨ [Start Simulation] ( ).
2. Select the checkbox of the programmable controller program data to be written on the "Online Data Operation" screen displayed, and click the [Execute] button. (Do not write "Simple Motion Module Setting" at this time.)
3. Close the screen once writing data has been completed.



The "SMM Simulator" screen is displayed.



4. Select [Online] ⇨ [Write to PLC] ( ) to display "Online Data Operation" screen again.
5. Select the checkbox of the "Simple Motion Module Setting" to be written on the "Online Data Operation" screen displayed, and click the [Execute] button.
6. When the "Write to Module" screen is displayed and the message to confirm whether or not to overwrite the contents in the flash ROM appears, click the [Yes] button.



7. Click the [RESET] button on the "GX Simulator3" screen.

The cooperative simulation is started.

11.3 External Device Operation Simulation

This section explains how to debug by using the I/O System Setting function.

I/O System Setting function

The I/O System Setting is a function that enables to debug without changing programs by simulating operations of input/output devices.

Precautions

- The simulator is in STOP state, the I/O System Setting function will not operate. When the state is changed STOP → RUN, it will operate from the beginning of I/O system setting data.
- Labels are obtained from a project that is open at the first execution of the I/O System Setting. If the labels are edited and written to a simulator after startup, they are not reflected. Close the "I/O System Setting" screen and start it again.

Execution procedure of I/O System Setting function

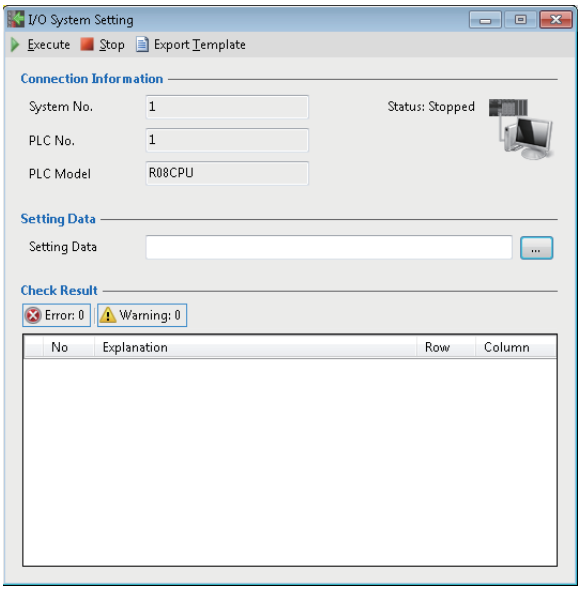
1. Start GX Simulator3. (☞ Page 227 Starting/Stopping Simulation)
2. Set GX Simulator3 to "STOP".
3. Display the "I/O System Setting" screen. (☞ Page 229 Executing I/O System Setting function)
4. Output the template file of the setting data. (☞ Page 230 Outputting the template of setting data)
5. Edit the outputted file. (☞ Page 230 Creating setting data)
6. Register the devices/labels to be monitored on the Watch window. (☞ Page 277 Checking Current Values by Registering Devices/Labels)
7. Open the edited file. (☞ Page 232 Opening setting data)
8. Execute the I/O system. (☞ Page 232 Executing the I/O system)
9. Set GX Simulator3 to "RUN".

Executing I/O System Setting function

Configure the devices to be used in the I/O System Setting, then execute the simulation.

Window

Select [Tool] ⇨ [I/O System Setting] on the "GX Simulator3" screen.



Outputting the template of setting data

Output the template of I/O system setting data in CSV format.

Operating procedure

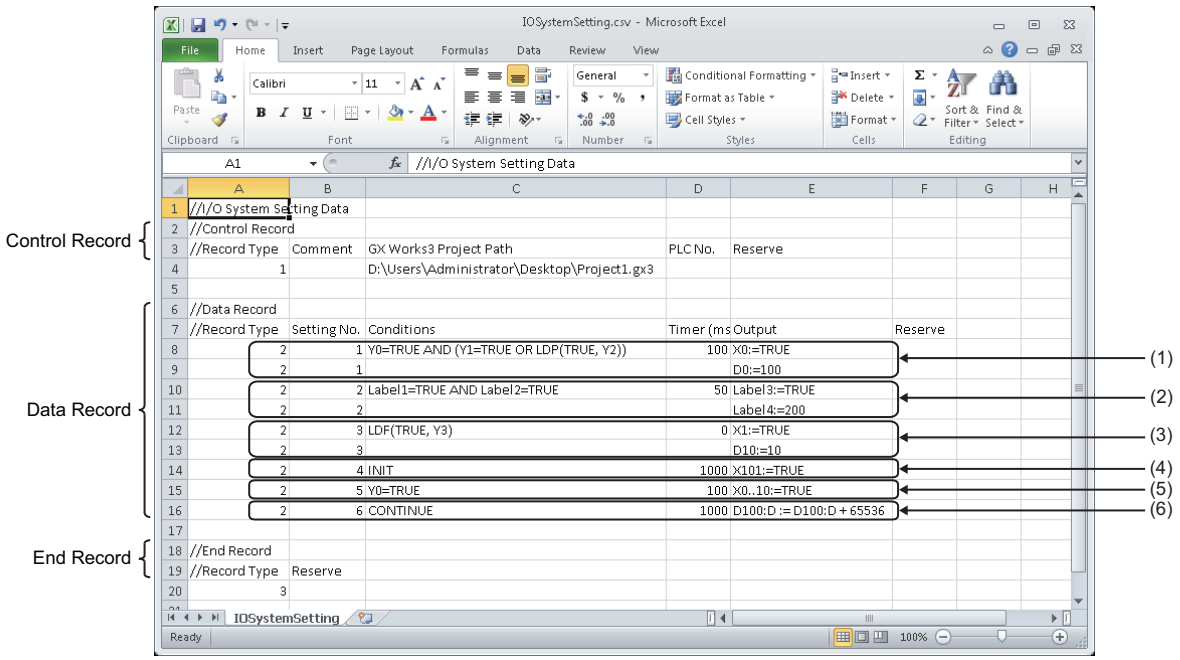
Select the [Export Template] on the "I/O System Setting" screen.

Creating setting data

Open the outputted template file, then edit the setting data.

For the usable devices/labels, refer to the following section.

☞ Page 233 Supported device/label



	A	B	C	D	E	F
1	//I/O System Setting Data					
2	//Control Record					
3	//Record Type	Comment	GX Works3 Project Path	PLC No.	Reserve	
4	1		D:\Users\Administrator\Desktop\Project1.gx3			
5						
6	//Data Record					
7	//Record Type	Setting No.	Conditions	Timer (ms	Output	Reserve
8	2	1	Y0=TRUE AND (Y1=TRUE OR LDP(TRUE, Y2))	100	X0:=TRUE	
9	2	1			D0:=100	
10	2	2	Label1=TRUE AND Label2=TRUE	50	Label3:=TRUE	
11	2	2			Label4:=200	
12	2	3	LDF(TRUE, Y3)	0	X1:=TRUE	
13	2	3			D10:=10	
14	2	4	INIT	1000	X101:=TRUE	
15	2	5	Y0=TRUE	100	X0..10:=TRUE	
16	2	6	CONTINUE	1000	D100:D := D100:D + 65536	
17						
18	//End Record					
19	//Record Type	Reserve				
20	3					

Displayed items

Item		Description
Control Record	Record Type (required)	'1' (the value indicates the control record) is displayed.
	Comment	Comments are set (up to 50 characters). (optional)
	GX Works3 Project Path (required)	Project path in the GX Works3 which is the start source of the simulator is displayed.
	PLC No.	Sets a PLC No. (1 to 8). (optional)
	Reserve	—
Data Record	Record Type (required)	'2' (the value indicates the data record) is displayed.
	Setting No. (required)	Specifies the setting number (1 to 1023) in ascending order. When specifying the multiple output on one condition, set the same number to each of them.
	Conditions (Only the first line in the setting number needs to be set.)	Specifies delimited conditions (up to 256 characters) in space. Up to six conditions can be combined in logical operation.
	Timer (ms) (Only the first line in the setting number needs to be set.)	Specifies the delay time (-1 to 1000) until the output is performed since the conditions have been satisfied. • -1: Output one per line for every scan when it is specified with the reserved word "CONTINUE".
	Output (required)	Specifies the output formula (up to 100 characters). When specifying the multiple output (up to 50) in one setting number, specify them in the next line.
	Reserve	—
End Record	Record Type (required)	'3' (the value indicates the end record) is displayed.
	Reserve	—

Ex.

Setting examples to CSV file

Pattern	Example	CSV file notation
(1) Basic device setting	When 'Y0' is turned ON and ('Y1' is turned ON or 'Y2' rises), 'X0' will be turned ON after 100ms. '100' is assigned to 'D0'	2,1,"Y0=TRUE AND (Y1=TRUE OR LDP(TRUE, Y2))",100,X0:=TRUE, 2,1,,,D0:=100,
(2) Label specification	When 'Label1' is turned ON and 'Label2' also is turned ON, 'Label3' will be turned ON after 50ms. '200' is assigned to 'Label4'.	2,2,Label1=TRUE AND Label2=TRUE,50,Label3:=TRUE, 2,2,,,Label4:=200,
(3) Instant specification	When 'Y3' falls, 'X1' will be turned ON immediately. '10' is assigned to 'D10'.	2,3,"LDF(TRUE, Y3)",0,X1:=TRUE, 2,3,,,D10:=10,
(4) After initialization specification	'X101' will be turned ON after 1s from execution.	2,4,INIT,1000,X101:=TRUE,
(5) Successive bit device output	When 'Y0' is turned ON, 'X0' to 'X10' will be turned ON after 100ms.	2,5,Y0=TRUE,100,X0..10:=TRUE,
(6) Timing chart 32-bit integer addition output	'65536' is added to 'D100' after 1s from the above pattern.	2,6,CONTINUE,1000,D100:D := D100:D + 65536,

Operating procedure

Open the outputted file in Excel® or the like and edit it directly.

Set the conditions by using the following character strings and symbols.

○: Available, ×: Not available

Category	String/ Symbol	Description	Availability				Usage example	
			Condition	Output	Device			
					Bit	Word		
Instruction	CONTINUE*1,*2	Keeps the previous condition	○	×	×	×	2,1,Y0=TRUE,1,X0:=TRUE 2,2,CONTINUE,1,X0:=FALSE 2,3,CONTINUE,1,LOOP	
	LOOP*1,*3	Repeats (It is used with "CONTINUE")	×	○	×	×		
	INIT*1,*4	Executes after initialization only one time	○	×	×	×	2,1,INIT,1,X0:=TRUE	
	LDP*5	LDP function	○	×	○	×	LDP(TRUE,Y0)	
	LDF*5	LDF function	○	×	○	×	LDF(TRUE,Y0)	
Operator	:=*6	Assigns	×	○	○	○	Bit	X0:=TRUE
							Word	D0:=1
	+	Adds	×	○	×	○	D0:=D0+1	
	-	Subtracts	×	○	×	○	D0:=D0-1	
	>*6,*7	Compares	○	×	×	○	D0>0	
	<*6,*7		○	×	×	○	D0<0	
	>=*6,*7		○	×	×	○	D0>=0	
	<=*6,*7		○	×	×	○	D0<=0	
	=*6,*7,*8		○	×	○	○	D0=0	
	<>*6,*7,*8		○	×	○	○	D0<>0	
	AND	Combines conditions using 'AND'	○	×	×	×	Y0=TRUE AND Y1=FALSE	
	OR	Combines conditions using 'OR'	○	×	×	×	Y0=TRUE OR Y1=FALSE	
Device type specifier*9	:U	Uses a device as a 16-bit non-negative integer	○	○	×	○	Condition	D0:U=0
							Output	D0:U:=0
	:D	Uses a device as a 32-bit integer	○	○	×	○	Condition	D0:D=0
							Output	D0:D:=0
	:UD	Uses a device as a 32-bit non-negative integer	○	○	×	○	Condition	D0:UD=0
							Output	D0:UD:=0
	:E	Uses a device as FLOAT [Single Precision]	○	○	×	○	Condition	D0:E=0
							Output	D0:E:=0
	:ED	Uses a device as FLOAT [Double Precision]	○	○	×	○	Condition	D0:ED=0
							Output	D0:ED:=0

Category	String/ Symbol	Description	Availability				Usage example	
			Condition	Output	Device			
					Bit	Word		
Symbol	..*1,*10	Specifies device range Example: X10..20	×	○	○	×	X10..20:=TRUE	
	(*11	Open bracket	○	×	○	○	Y0=TRUE AND (Y1=TRUE OR Y2=FALSE)	
	)*11	Close bracket	○	×	○	○		
Constant	TRUE*12	Turns ON a bit device	○	○	○	×	Condition	Y0=TRUE
							Output	X0:=TRUE
	FALSE	Turns OFF a bit device	○	○	○	×	Condition	Y0=FALSE
							Output	X0:=FALSE

*1 These are the original description methods in the I/O System Setting.
*2 An error occurs if "CONTINUE" is written at the top of a data record.
*3 When it is used without combining with "CONTINUE", error does not occur but "LOOP" is ignored.
An error occurs if "LOOP" is written at the top of a data record.
*4 The timing that the execution transition of "INIT" is enabled are as follow:
· The I/O System Setting was executed while the simulator is in RUN state
· The simulator was switched to RUN state while the I/O System Setting is executing
*5 Use both instructions according to the Structured Text notation.
MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)
*6 For devices without a type specifier, it can perform the operations with both constants Word [Signed] and Word [Unsigned]. In that case, device type is determined according to the constant type.
*7 To compare devices or labels each other, data types need to be consistent.
*8 To compare with equal/inequal in bit type, device and constant requires to be a set.
*9 For the devices that can be added suffixes (type specifier), refer to the following manual.
MELSEC iQ-R Structured Text (ST) Programming Guide Book
*10 'A' to 'F' can be entered to the device numbers written in hexadecimal format. An error will occur if the numbers are specified in descending order such as 'X10..2'.
*11 There is no limit to the number of brackets if the combination of open brackets and close brackets matches.
*12 Only bit devises such as 'Y0' can be used on conditions. (In this case, it indicates "Y0=TRUE".)

Precautions

The row beginning with '//' is handled as a comment, and it is skipped.

Opening setting data

Open a file of edited setting data.

Operating procedure

1. Click the [...] button in the setting data column on the "I/O System Setting" screen.
2. Select a target file, then click the [Open] button.

Precautions

Language-specific characters are included in the files of setting data.
When the language font to be displayed is not installed in the personal computer, some characters may be garbled.

Executing the I/O system

Execute the I/O system after checking the I/O System Setting Data.

Operating procedure

Select [Execute] on the "I/O System Setting" screen.
When the check result includes an error, the description will be displayed on the result screen. Correct the I/O system setting data, then execute the system again.

Stopping the I/O system

Operating procedure

Select [Stop] on the "I/O System Setting" screen.

Supported device/label

Device

Local device is not supported.

○: Supported, ×: Unsupported

Category	Device name	Symbol		Digit specification	Bit-specified word device
User device	Input	X		○	×
	Output	Y		○	×
	Internal relay	M		○	×
	Latch relay	L		○	×
	Link relay	B		○	×
	Annunciator	F		○	×
	Link special relay	SB		○	×
	Timer	T	TS	×	×
			TN	×	×
			T	×	×
	Retentive timer	ST	STS	×	×
			STN	×	×
			ST	×	×
	Long timer	LT	LTS	×	×
			LTN	×	×
			LT	×	×
	Long Retentive Timer	LST	LSTS	×	×
			LSTN	×	×
			LST	×	×
	Counter	C	CS	×	×
			CN	×	×
			C	×	×
	Long counter	LC	LCS	×	×
			LCN	×	×
			LC	×	×
	Data register	D		×	○
	Link register	W		×	○
	Link special register	SW		×	○
System device	Special relay	SM		○	×
	Special register	SD		×	○
Link direct device (J□\□)	Link input	X		○	×
	Link output	Y		○	×
	Link relay	B		○	×
	Link special relay	SB		○	×
	Link register	W		×	○
	Link special register	SW		×	○
Module access device (U□\G□)	Module access device	G		×	○
CPU buffer memory access device (U3E□\G□/HG□)	CPU buffer memory access device	G/HG		×	○

Category	Device name	Symbol	Digit specification	Bit-specified word device
File register	File register	R	×	○
		ZR	×	○
Module refresh register	Module refresh register	RD	×	○
Constant	Decimal constant	K	×	×
	Hexadecimal constant	H	×	×
	Real constant	E	×	×

Label

The following labels are supported.

- Global label, Module label
- Labels of which class is "VAR_GLOBAL"/"VAR_GLOBAL_RETAIN"
- Labels except for string (32)/string [Unicode] (32)/pointer type

Precautions

- Array type/structure type labels, which were assigned labels by all conversion, are not supported.
- The variable of function block cannot be specified.
- Labels are not supported when the device of assignment target is not supported or the assignment target is a label.
- Timer/retentive timer/counter labels need to specify contacts/coils/current value as same as devices.
- Devices/labels cannot used to specify elements of array.

12 SETTING ROUTE TO CPU MODULE

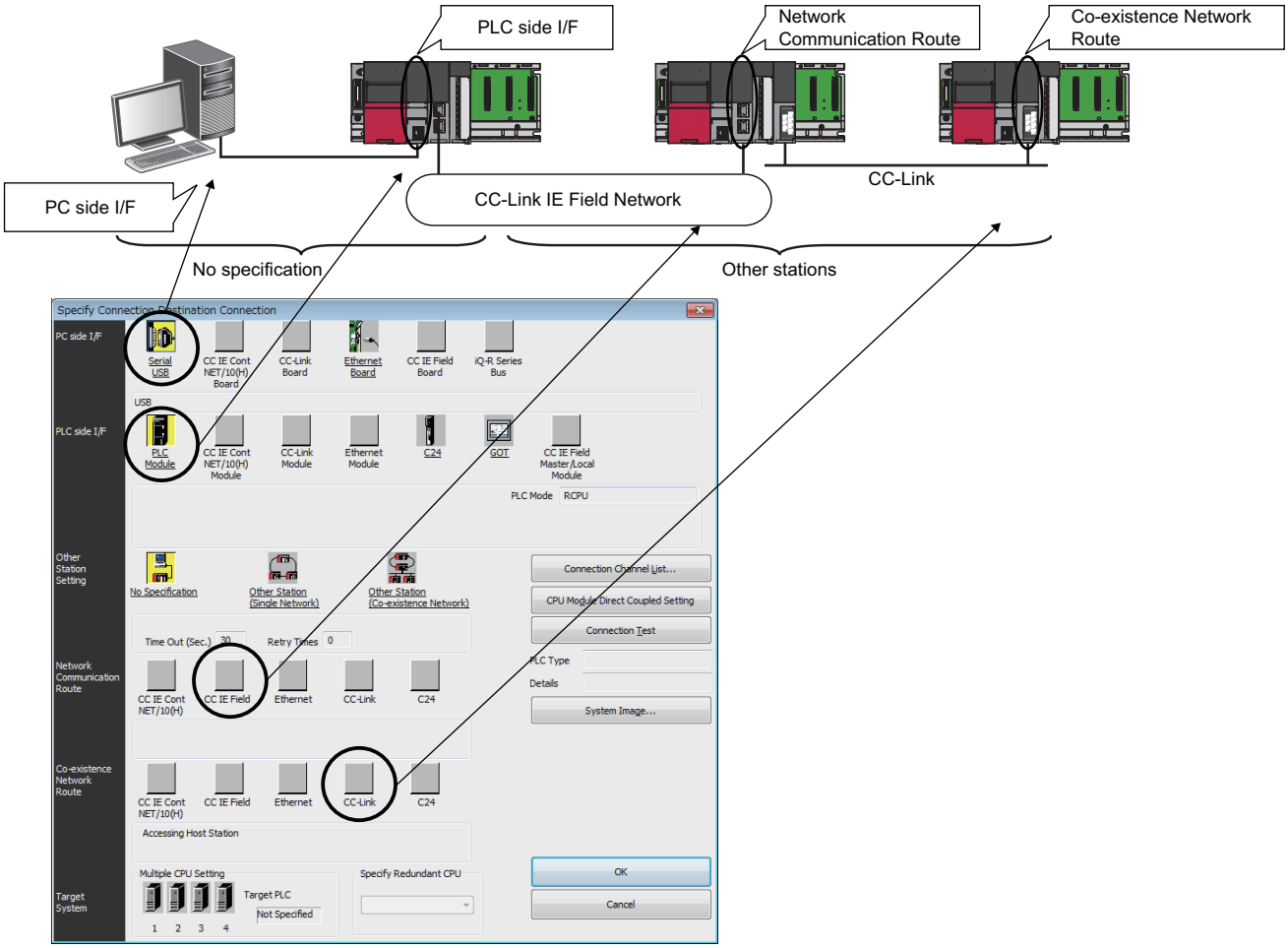
This chapter explains how to set the connection destination for accessing a CPU module from GX Works3.

12.1 Specification of Connection Destination

This section explains how to set communication routes, including the interfaces both the personal computer and CPU module, and the routing networks, for accessing a CPU module on the "Specify Connection Destination" screen.
The module type of a project must be set to that of the CPU to be accessed.

Window

[Online] ⇨ [Specify Connection Destination]



- Double-click the items underlined on the screen to set the details of each item.
- The icons colored yellow indicate that the settings have been applied.
- For FX5CPU, the icons of the unsettable connection destination route are not displayed.

Displayed items

Item		Description
Other Station Setting	No Specification	Specify this to access the CPU module directly connected to a personal computer.
	Other Station (Single Network) *1	Specify this to access the CPU module on other station via only one kind of network (including a multi-tier system). Since Ethernet is regarded as equivalent to CC-Link IE Controller Network, CC-Link IE Field Network, and MELSECNET/10(H), specify "Single Network" for a system in which those are mixed.
	Other Station (Co-existence Network) *1	Specify this to access the CPU module on other station via two kinds of network.

Item	Description
Network Communication Route	Select the network type, network number, station number, and start I/O number of the network that is routed for accessing the CPU module on other station. The setting items differ depending on the selected network type.
Co-existence Network Route	Select the network type, network number, station number, and start I/O number of the network to access. The setting items differ depending on the selected network type.
Target System	Specify the access target in the multiple CPU system.

*1 To specify the own station, select "No Specification".

Display the illustration of the connection route

Click the [System Image] button to display the set connection route in an illustration to check the route.

12.2 Direct Connection

This section explains the setting method for accessing the CPU module directly connected to a personal computer.

Point

Click the [CPU Module Direct Coupled Setting] button on the "Specify Connection Destination" screen to change the settings to direct connection.

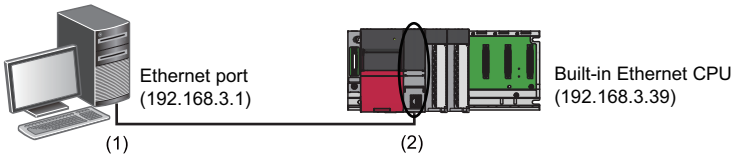
USB connection

The following shows the setting example for accessing the CPU module on the own station with USB from GX Works3.
FX5CPU does not support this connection.

No.	Item	Item to be selected	Internal setting	Input value
(1)	PC side I/F	Serial/USB	USB	—
(2)	PLC side I/F	PLC Module	PLC Mode	RCPU
	Other Station Setting	No Specification	Check at Communication Time	30 seconds
			Retry Times	0 time

Ethernet connection

The following shows the setting example for accessing the built-in Ethernet CPU with Ethernet from GX Works3.



No.	Item	Item to be selected	Internal setting	Input value
(1)	PC side I/F	Ethernet Board	Network No.*1	—
			Station No.*1	—
			Protocol*1	TCP
(2)	PLC side I/F	PLC Module	Ethernet Port Direct Connection	—
	Other Station Setting	No Specification	Check at Communication Time	30 seconds
			Retry Times	0 time

*1 For FX5CPU, these setting items are not available.

Point

For FX5CPU, the Ethernet adapter at the personal computer side, which is used for the Ethernet port direct connection, can be specified.
Click the [CPU Module Direct Coupled Setting] button on the "Specify Connection Destination" screen, and select an adapter displayed on the "CPU Module Direct Coupled Setting" screen.

Precautions

■Windows Vista® or later

A warning message may appear.
Click the [Unblock] button for Windows Vista®, and click the [Allow access] button for Windows® 7 or later to unblock the program (permit the access) and continue the operation.

■When the same IP addresses are displayed

When the [Find] button is clicked on the "PLC side I/F Detailed Setting of PLC Module" screen, the same IP addresses may be displayed in the list of the connection destination CPU.

Two or more IP addresses may be set to the [IP Settings] tab on the "Advanced TCP/IP Settings" screen of Windows® network settings. Set only one IP address.

■When the personal computer does not allow to communicate with GX Works3 by the Windows® Firewall

When the Windows ® Firewall is enabled and not allowed to communicate with GX Works3, a timeout may occur.

To allow the communication, set the personal computer as shown below.

When using other software with a firewall function, refer to the manual of the software and allow the communication with GX Works3.

Ex.

For Windows Vista®

1. Select [Control Panel] ⇨ [Security] ⇨ [Windows Firewall] ⇨ [Allow a program through Windows Firewall] from the Start menu of Windows®.
2. Click the [Add Program] button in the [Exception] tab.
3. Select "GX Works3" on the "Add a Program" screen, and click the [OK] button.
4. Select "GX Works3"*1 in the list, and click the [OK] button.

*1 'gxw3' may be displayed when RCPU/Ethernet module is searched on the network and access through the Windows® firewall is set to allow before Ethernet port direct connection.

Ex.

For Windows® 7 or later

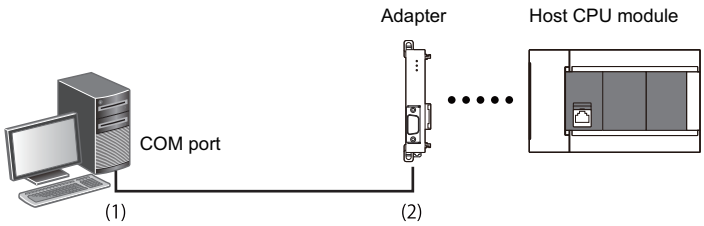
1. Select [Control Panel] ⇨ [System and Security] ⇨ [Windows Firewall] ⇨ [Allow a program*1 or feature through Windows Firewall] from Windows® Start.
2. Click the [Change settings] button, and click the [Allow another program*1] button.
3. Select "GX Works3" on the "Add a Program*1" screen, and click the [Add] button.
4. Select "Domain", "Home/Work (Private)", and "Public" of the added GX Works3*2 in the list, and click the [OK] button.

*1 For Windows® 8 or later, "Apps" is displayed.

*2 'gxw3' may be displayed when RCPU/Ethernet module is searched on the network and access through the Windows® firewall is set to allow before Ethernet port direct connection.

Serial connection

The following shows the setting example for accessing a CPU module on the own station via a serial port from GX Works3.
RCPU does not support this connection.

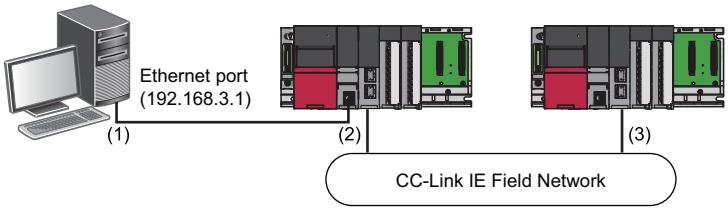


No.	Item	Item to be selected	Internal setting	Input value
(1)	PC side I/F	Serial/USB	RS-232C	—
			COM port	COM1
			Transmission Speed	115.2Kbps
(2)	PLC side I/F	PLC Module	PLC Mode	FX5CPU
	Other Station Setting	No Specification	Check at Communication Time	30 seconds
			Retry Times	0 time

12.3 Accessing via Network (Single Network)

This section shows the setting example for accessing a CPU module on other station via single network after accessing the Ethernet built-in CPU from GX Works3.

FX5CPU does not support it.



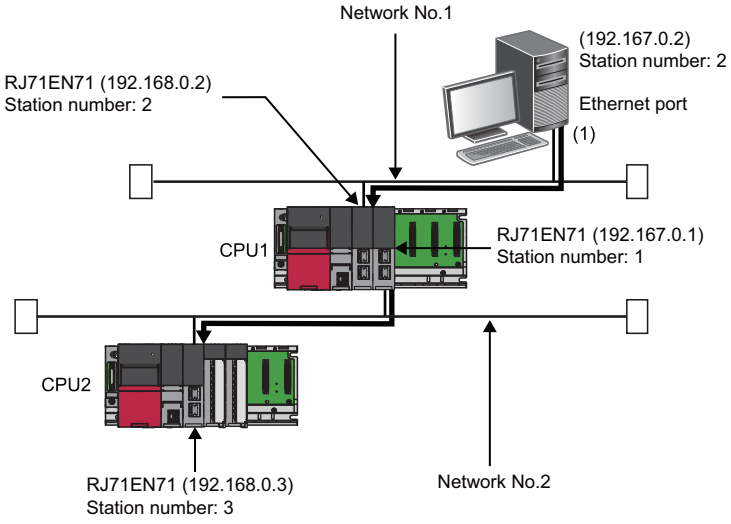
No.	Item	Item to be selected	Internal setting	Input value
(1)	PC side I/F	Ethernet Board	Network No.	—
			Station No.	—
			Protocol	TCP
(2)	PLC side I/F	PLC Module	Ethernet Port Direct Connection	—
	Other Station Setting	Other Station (Single Network)	Check at Communication Time	30 seconds
			Retry Times	0 time
(3)	Network Communication Route	CC IE Field	Network No.	1
			Station No.	0

Access via Ethernet module

The following shows the setting example for accessing the CPU module via CC-Link IE built-in Ethernet module from GX Works3.

It is same settings when accessing a CPU module via the network part of RnENCPU.

FX5CPU does not support it.



No.	Item	Item to be selected	Internal setting	Input value
(1)	PC side I/F	Ethernet Board	Network No.	1
			Station No.	2
			Protocol	TCP

No.	Item	Item to be selected	Internal setting		Input value
(2)	PLC side I/F	Ethernet Module	PLC Type		RJ71EN71
			Network No.		—
			Station No.		1
			IP Address		192.167.0.1
			IP Input Format		DEC
			Station No. <-> IP Information		Automatic Response System
	Other Station Setting	Other Station (Single Network)	Check at Communication Time		30 seconds
			Retry Times		0 time
(3)	Network Communication Route	Ethernet	Access to other station in the same loop or multi-tier system	Network No.	2
				Station No.	3

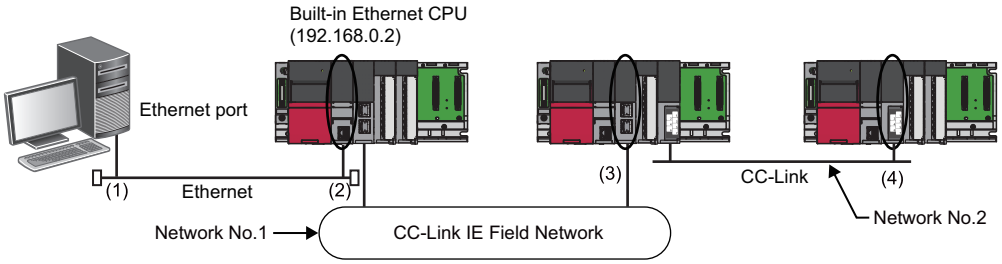
Precautions

- When communicating with multiple GX Works3s, use TCP/IP or UDP/IP communication.
- For a multiple network system, setting the network dynamic routing is required.

MELSEC iQ-R Ethernet User's Manual (Application)

12.4 Accessing via Network (Co-existence Network)

The following shows the setting example for accessing the CPU module via co-existence network from a personal computer. FX5CPU does not support it.



No.	Item	Item to be selected	Internal setting		Input value
(1)	PC side I/F	Ethernet Board	Network No.		1
			Station No.		1
			Protocol		TCP
(2)	PLC side I/F	PLC Module	Connection via HUB	IP Address	192.168.0.2
				Response Wait Time	2 seconds
	Other Station Setting	Other Station (Co-existence Network)	Check at Communication Time		30 seconds
			Retry Times		0 time
(3)	Network Communication Route	CC IE Field	Network No.		1
			Station No.		0
(4)	Co-existence Network Route	CC-Link	Start I/O		20
			Station No.		1

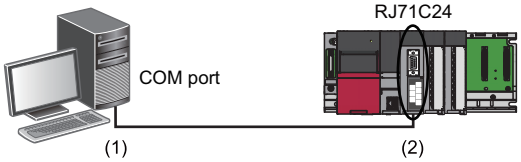
12.5 Accessing via Serial Communication Module

This section explains the setting method for accessing the CPU module on the own station or on other station via a serial communication module.

FX5CPU does not support it.

Connection on a 1:1 basis

The following shows the setting example for accessing the CPU module by connecting a personal computer and the serial communication module.



No.	Item	Item to be selected	Internal setting	Input value
(1)	PC side I/F	Serial/USB	COM port	COM1
			Transmission Speed	115.2Kbps
(2)	PLC side I/F	C24	PLC Type	RJ71C24
			Station No.*1	—
			Parity	—
			Sum Check	—
	Other Station Setting	No Specification	Check at Communication Time	—
			Retry Times	—

*1 Set the same station number as RJ71C24.
Set the station number on the Module Parameter screen.

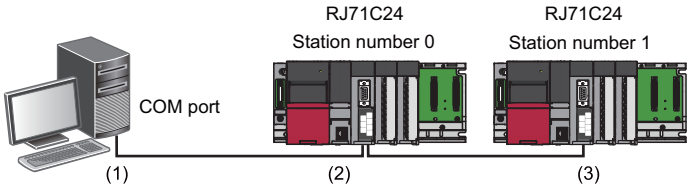
Connection on a 1:n basis

The following explains how to access the CPU module on other station from GX Works3 in a system composed of multiple CPU module stations.

Access via a serial communication module

The following shows the setting example for accessing the CPU module on other station via serial communication module. When connecting via serial communication module, the module parameter for MELSOFT connection is required. For details, refer to the following manual.

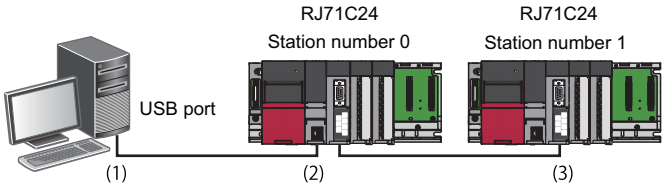
MELSEC iQ-R Serial Communication Module User's Manual (Application)



No.	Item	Item to be selected	Internal setting	Input value
(1)	PC side I/F	Serial/USB	COM port	COM1
			Transmission Speed	115.2Kbps
(2)	PLC side I/F	C24	PLC Type	RJ71C24
			Station No.	0
			Parity	Odd
			Sum Check	—
	Other Station Setting	Other Station (Single Network)	Check at Communication Time	30 seconds
			Retry Times	0 time
(3)	Network Communication Route	C24	Start I/O	20
			Station No.	1

Direct connection of a CPU module

The following shows the setting example for accessing the CPU module on other station via serial communication module by connecting a personal computer and CPU module directly.



No.	Item	Item to be selected	Internal setting	Input value
(1)	PC side I/F	Serial/USB	USB	—
(2)	PLC side I/F	PLC Module	PLC Mode	RCPU
	Other Station Setting	Other Station (Single Network)	Check at Communication Time	30 seconds
			Retry Times	0 time
(3)	Network Communication Route	C24	Start I/O	20
			Station No.	1

12.6 Accessing via GOT (GOT Transparent Function)

This section explains how to access the CPU module from GX Works3 via GOT using the GOT transparent function.

Precautions

■Considerations when performing online operations from GX Works3

Do not perform online operations from GT Designer2 or GT Designer3 to GOT (such as downloading project data) when online operations are being performed from GX Works3 to the CPU module using the GOT transparent function.

■When GOT does not monitor normally

The GOT transparent function cannot be used in the following cases.

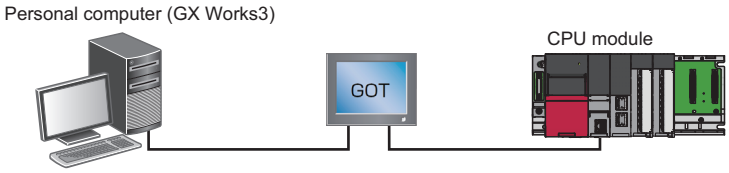
- When GOT does not monitor normally due to CPU module errors or communication errors between the CPU module and GOT
- During the period of time between turning ON or resetting the CPU module or GOT and the start of GOT monitoring

Check the following items if GOT does not monitor normally.

Item	Reference
Does the CPU module operates normally?	☞ Page 314 Diagnosing CPU Module
Is the CPU module connected to GOT normally?	🕮 Manual of GOT used

Accessing via GOT

The following shows the setting for accessing the CPU module via GOT.



The availability of connection differs depending on the GOT series and the connection condition between the personal computer and GOT.

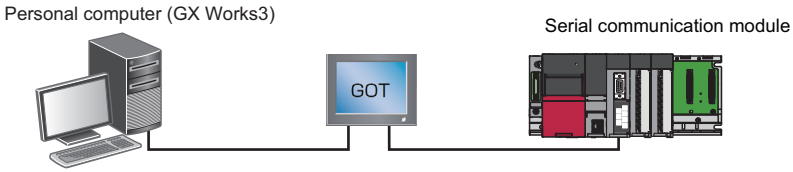
For details, refer to the following manuals.

🕮 GOT2000 Series Connection Manual (Mitsubishi Products) For GT Works3 Version1

🕮 GOT1000 Series Connection Manual (Mitsubishi Products) for GT Works3

Accessing via module

The following shows the setting of GX Works3 for accessing the CPU module via GOT and serial communication module. FX5CPU does not support it.



The availability of connection differs depending on the GOT series and the connection condition between the personal computer and GOT.

For details, refer to the following manuals.

🕮 GOT2000 Series Connection Manual (Mitsubishi Products) For GT Works3 Version1

🕮 GOT1000 Series Connection Manual (Mitsubishi Products) for GT Works3

12.7 Accessing to Multiple CPU System

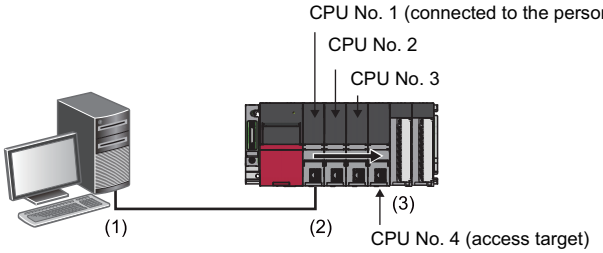
This section explains the setting method for accessing the CPU module (host CPU) directly connected to the personal computer or other CPU modules (other CPUs) in the multiple CPU system.
The setting method for accessing the multiple CPU system on other station via a network is also explained.
FX5CPU does not support it.

Accessing host CPU

The setting for accessing the host CPU is the same as that for accessing the CPU module on the own station. (☞ Page 237 Direct Connection)

Accessing other CPUs

In the multiple CPU system, specify the CPU number (PLC No.1 to 4) of the access target at "Multiple CPU Setting" when accessing a CPU module that is not directly connected to the personal computer.
The following shows the setting for accessing the CPU No.4 by connecting a personal computer to CPU No.1.



No.	Item	Item to be selected	Internal setting	Input value
(1)	PC side I/F	Serial/USB	USB	—
(2)	PLC side I/F	PLC Module	PLC Mode	RCPU
	Other Station Setting	No Specification	Check at Communication Time	30 seconds
			Retry Times	0 time
(3)	Target System	Multiple CPU Setting	Target PLC	PLC No.4

Accessing via network

The following explains the setting methods for accessing the CPU module in a multiple CPU system on other station via a network.

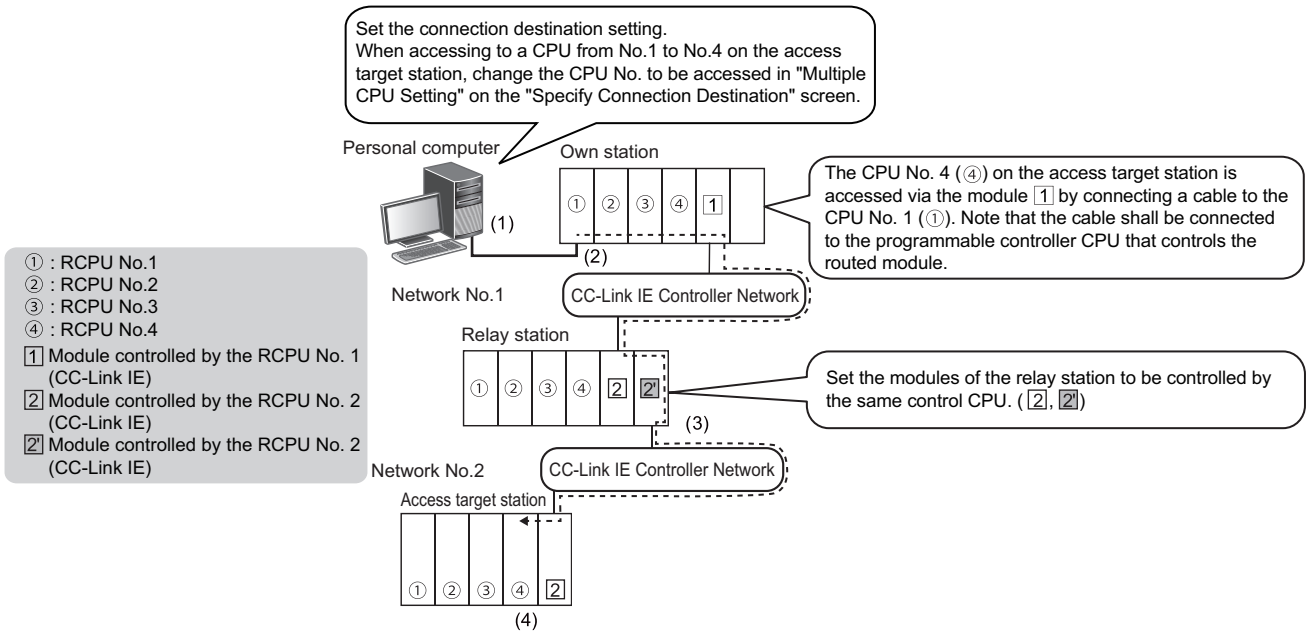
■Relay station

Set the parameters so that the module mounted on the relay station is controlled by the same CPU module. (In the figure below, PLCNo. 2 is the control CPU.)

■Access target station

Set the PLC number in "Multiple CPU Setting" on the "Specify Connection Destination" screen when the access target station is the multiple CPU system.

The following shows the setting for accessing the PLC No.4 in the multiple CPU system on the access target station via a network.


Set the connection destination setting.
When accessing to a CPU from No.1 to No.4 on the access target station, change the CPU No. to be accessed in "Multiple CPU Setting" on the "Specify Connection Destination" screen.
Personal computer
Own station
(1)
(2)
The CPU No. 4 (④) on the access target station is accessed via the module 1 by connecting a cable to the CPU No. 1 (①). Note that the cable shall be connected to the programmable controller CPU that controls the routed module.
① : RCPU No.1
② : RCPU No.2
③ : RCPU No.3
④ : RCPU No.4
1 Module controlled by the RCPU No. 1 (CC-Link IE)
2 Module controlled by the RCPU No. 2 (CC-Link IE)
2' Module controlled by the RCPU No. 2 (CC-Link IE)
Network No.1
CC-Link IE Controller Network
Relay station
Set the modules of the relay station to be controlled by the same control CPU. (2, 2')
(3)
Network No.2
CC-Link IE Controller Network
Access target station
(4)

No.	Item	Item to be selected	Internal setting	Input value
(1)	PC side I/F	Serial/USB	USB	—
(2)	PLC side I/F	PLC Module	PLC Mode	RCPU
	Other Station Setting	Other Station (Single Network)	Check at Communication Time	30 seconds
			Retry Times	0 time
(3)	Network Communication Route	CC IE Cont NET/10(H)	Network No.	2
			Station No.	0
(4)	Target System	Multiple CPU Setting	Target PLC	PLC No.4

12.8 Considerations of Communication with CPU module

This section explains the considerations of communication with the CPU module.
Refer to the considerations described in each section as well as the descriptions in this section.

Considerations of communication via multiple network systems

When accessing RCPU via multiple network systems, all the connected station and relay stations must be RCPU.
Connected station: Station directly connected from GX Works3
Relay station: Stations to be routed on network system

Considerations of communication with the CPU module using a USB cable

■Connect and disconnect a USB cable, reset the CPU module, and turn the power ON/OFF

A communication error may occur and it may not be recovered if connecting and disconnecting a USB cable, resetting the CPU module, or turning the power ON/OFF is performed frequently during communication with the CPU module.
Be sure to set GX Works3 to offline status as much as possible during these operations. 'Offline' indicates the state other than below.

- Write to PLC/Read from PLC, Monitor, and PLC diagnostics

If the operation is not recovered from an error, remove the USB cable. Then, connect it again after five or more seconds.
(Even after this operation, an error may occur at initial communication. However, communication will be successful after that.)
Click the [OK] button on the warning message, and remove a USB cable from the personal computer.

■Combination of personal computers and USB cables

A communication error may occur depending on the combination of personal computers and USB cables.
If an error occurs, take appropriate actions in accordance with the message displayed on the screen.

Other considerations

■High-speed communication using an RS-232 cable

Communication may fail depending on the performance of the personal computer when high-speed communication is attempted by changing the transmission speed at the serial port of the personal computer (personal computer side interface).
Communication speed may also slow down due to communication retries.
Decrease the transmission speed if high-speed communication cannot be performed normally.

■Resume function, suspend setting, power saving function, and standby mode of the personal computer

A communication error may occur during communication with the CPU module when any of the settings shown above are enabled.
Disable these settings for communication with the CPU module.

13 WRITING/READING DATA TO CPU MODULE

This section explains the operations for writing/reading data to/from a CPU module or memory card, and verifying projects.
For the methods for writing the clock setting to a CPU module, refer to the following section.
☞ Page 325 Setting Clock in CPU Module

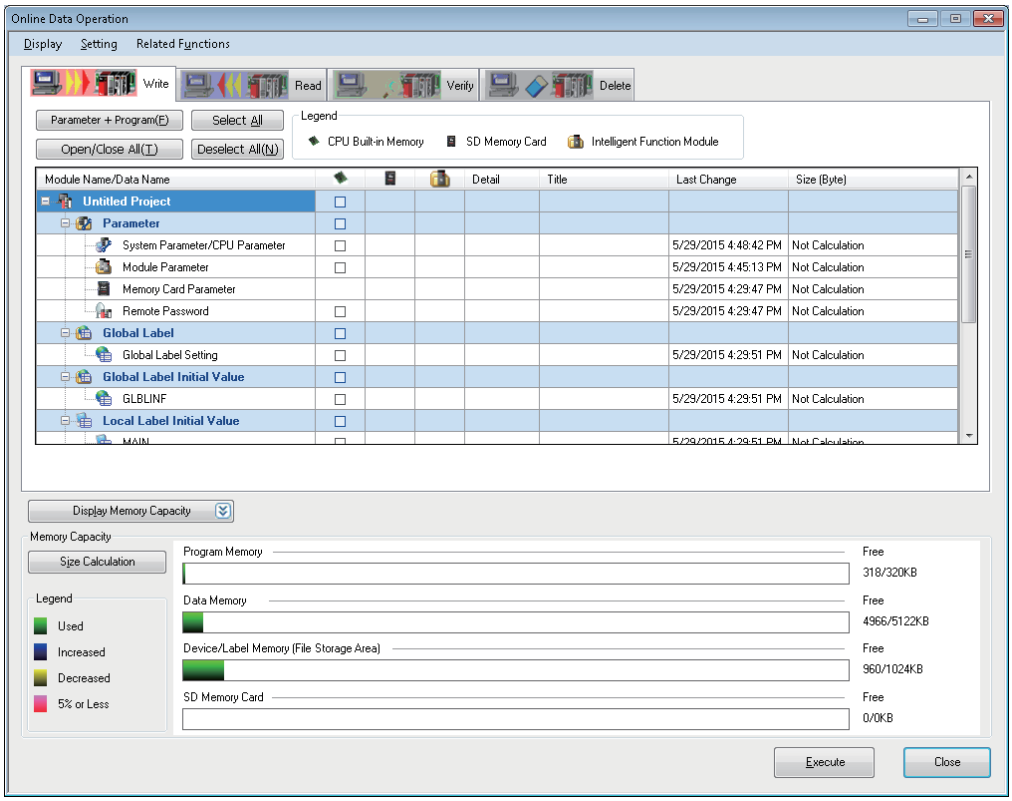
13.1 Writing/Reading Programmable Controller Data

Write the created data to/read from a CPU module or a memory card on the "Online Data Operation" screen.

Configuration of Online Data Operation screen

Window

[Online] ⇨ [Write to PLC] ( )/[Read from PLC] ( )/[Delete PLC Data]
The example of writing data is shown below.



- The data name displayed in gray indicates that it is in the unconverted state.
- When a data is written to the programmable controller, the target memory capacity may be displayed smaller than the actual file size.
- Update the graph of the memory capacity by clicking the [Size Calculation] button.

Precautions

■Memory capacity display for FX5CPU

- Program memory (The capacity of program memory is displayed in unit of step.)
- Data memory (The program, restored information, parameters, and device comments are displayed separately.)
- SD memory card (The capacity of the entire SD memory card is displayed.)
- SD memory card (The program, restored information, parameters, and device comments are displayed separately.)

■Safety project and safety operation mode of RnSFCPU

The safety data and device memory of a safety project can only be written when the safety operation mode of RnSFCPU is in the test mode. If the module is in the safety mode, switch the mode to the test mode.

☞ Page 327 Switch Safety Operation Mode

■Data to be written and write target

○: Writable, ×: Not writable

Data name		Write target		
		CPU built-in memory	SD memory card	Intelligent function module
System Parameter/CPU Parameter		○	○	×
Module Parameter		○	○	×
Module Extended Parameter		○	○	○
Memory Card Parameter		×	○	×
Remote Password		○	○	×
Global Label	Global Label Setting	○	○	×
	Global Label Assignment Information*1	○	○	×
Program File		○	○	×
FB File/FUN File		○	○	×
Label Initial Value File*1		○	○	×
Device Memory (File Register)		○	×	×
Device Initial Value		○	○	×
Device Comment*2		○	○	×

*1 Not supported by FX5CPU.

*2 FX5CPU only supports common device comments and does not support each program device comments.

For the details on the availability of writing safety data, refer to the following manual.

📖 MELSEC iQ-R CPU Module User's Manual (Application)

■How to check the data size to be written

The data size to be written can be displayed on the "Size (Byte)" column by enabling the file size display.

Select [View] ⇨ [Display File Size] on the "Online Data Operation" screen to enable the file size display.

■Display of memory capacity

The display content differs depending on the setting of file size display.

Disabled: The current memory capacity of the write target is displayed.

Enabled: The memory capacity, to which the data size to be written/deleted has been reflected, is displayed.

Writing data to programmable controller

Write data to a CPU module.

For the considerations for writing, refer to the following section.

☞ Page 256 Considerations for Online data operation

Operating procedure

1. Select the [Write] tab on the "Online Data Operation" screen.
2. Select a file to be written and a write target.
3. Click the [Detail] button to set the details such as a range to be written.
4. Click the [Execute] button.

After writing a safety program or a parameter to an RnSFCPU, manuals are displayed to clarify the information that requires to be confirmed by users for using a safety project safely.

Point

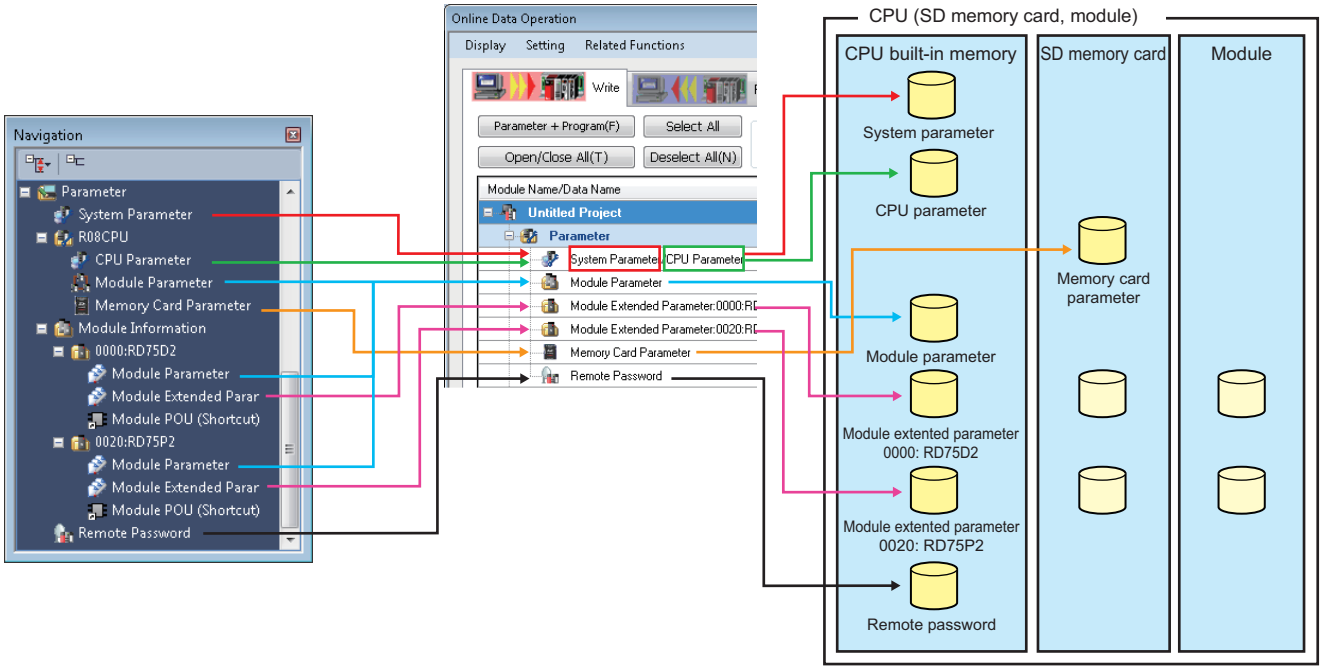
When more than one program is written to the CPU module, the programs categorized as "No Execution Type" will never be executed on the CPU module. For saving the memory capacity of the CPU module, removing them from the target data at writing data is recommended.

Parameter

Write "Parameter" in accordance with the operation or change of the system.

■Relation between module parameter on the Project view and module parameter in CPU module

Module parameters are written with all other parameters displayed on the Project view in one file.



Therefore, when module parameters exist in the CPU module, the parameter file is overwritten.

Ex.

When writing four module parameters to the CPU module in which module parameters of five modules exist; the settings of five modules are overwritten with the settings of four modules, and one setting that does not exist in the project will be deleted.

■Consistency of parameters

When writing a part of the parameters additionally or writing parameters after changing the parameters, the consistency of the parameters are required. Consistency means making the information of I/O Assignment Setting (start I/O number, slot number, etc.) between the system parameters and each parameter match.

There are two methods to keep the consistency.

- Read the system parameter and each parameter to change them. After that, write them to a CPU module. Resetting the CPU module is required to change the system parameter.
- Read each parameter to be changed, and change them to be consistent with the system parameter in the project. After that, write the changed parameters only to the CPU module. Basically, resetting CPU module is not required, however, some modules are required to reset the CPU module after changing parameters. For details, refer to the manual of the module used.

■Modules with start I/O number unset

The unset module parameters cannot be written to a programmable controller.

■Writing parameters of remote I/O module (slave station) on CC-Link IE Field Network

Write the parameters for slave station on the CC IE Field Configuration window. For details, refer to the manual of each remote I/O module.

Programs

After creating a program (local label, program), write "Program".

When the column of "Initial Value" is set on the local label editor, write "Local Label Initial Value".

FX5CPU does not support initial value of label.

When a standard function or a standard function block is used in the program, write "SlibFbFile" of POU.

■Setting the secured steps for online program change

Set the range of program (program file) to be written and steps for online program change to a CPU module on the "Program Detail Setting" screen displayed by clicking the [Detailed] button of "Program".

FX5CPU does not support this setting.

Item	Description
Allocating memory for online program change	Writing programs while the CPU module is in RUN affects scan time due to the change of number of steps. Enter the secured steps for online program change to handle the change of number of steps when performing the online program change function. When the number of program steps is changed, the capacity of program files is not changed if the changed number of steps are within the range of the number of the secured steps for online program change.

■Operations after writing SFC programs

For SFC programs, the operations performs differently between Ladder, ST, and FBD/LD when changing programs by writing them to a programmable controller. For the details, refer to the following manual.

MELSEC iQ-R Programming Manual (Program Design)

Global label

After setting global labels, write "Global Label Setting".

When the column of "Initial Value" is set in the global label editor, write "Global Label Initial Value".

When "Access from External Device" is selected on the global label editor, write "Global Label Assignment Information".

FX5CPU does not support the initial value of labels nor the setting for accessing from external device.

Precautions

The sample comments are included in the write target. If the data size to be written exceeds the memory capacity of the CPU module at writing data, prepare an SD memory card.

For FX5CPU, the writable capacity of the SD memory card is the same as that of CPU module.

Device memory

When writing values to the device memory area of the CPU module after setting device memory, write "Device Memory".
The device memory which has the same name as the program is written to a local device.
In contrast, the device memory which has the different name as the program is written to a global device.
Only one device memory for a global device can be selected.

Precautions

FX5CPU does not support local devices.
The device memory is written to a global device regardless of whether or not it is consistent with the program name.

■Setting write target devices and their ranges

Set the type and range of device memory to be written to the CPU module on the "Device Data Detail Setting" screen displayed by clicking the [Detailed] button of "Device Memory".
Devices that can be written are as follows:
○: Writable, ×: Not writable, —: No corresponding device

Device	Global device	Local device*1
M, V*1, T, ST, C, D, R*2, SA\M, SA\T, SA\ST, SA\C, SA\D	○	○
L, B, F, SB, LT*1, LST*1, LC, W, SW, Z, LZ, RD*1, SA\B, SA\W	○	—
X, Y, S, SD, SM, SA\X, SA\Y, SA\SD, SA\SM	×	—

*1 Not supported by FX5CPU.
*2 Not supported by RCPU.

File register

When writing the device memory which has been set as a file register, write "File Register".
FX5CPU does not support this setting.

■Setting range to be written

Set the range of file register to be written to the CPU module on the "File Register Detail Setting" screen displayed by clicking the [Detailed] button of "File Register".
If two or more file registers are selected, the value of each file register is written/read in the same range.

Precautions

Before writing file registers, write the CPU parameters, then turn the power of the CPU module OFF → ON/reset the CPU module.

Device initial values

After setting device initial values, write "Device Initial Value".
The device initial value which has the same name as the program is written as a file for the local device. The devices which cannot be used as a local device cannot be written.
The device initial values which have difference name as the program will be written as a file for a global device.
FX5CPU does not support local devices.
The device initial values are written to a global device regardless of whether or not it is consistent with the program name.

Device comment

After setting device comments, write "Common Device Comment" or "Each Program Device Comment".
For FX5CPU, each program device comment cannot be written to CPU module.

■Setting range to be written

Set the range of "Common Device Comment"/"Each Program Device Comment" to be written to a CPU module on the "Device Comment Detail Setting" screen displayed by clicking the [Detailed] button of "Common Device Comment" or "Each Program Device Comment".
Set this setting to limit the writing range of device comments in cases such as when the CPU module has small free capacity.
When the range of device comments is not set, all device comments are written.

Item	Description
Number of characters per 1 comment	Set the maximum number of comments to be written to a CPU module. When the character length of an input value is less than the number of characters of device comment set for the project, the device comment with less number of characters is written to a CPU module. Therefore, mismatched data may be detected when verifying project data.

Precautions

When the sample comments are read to device comments, all the sample comments are written to the CPU module. Therefore, the data size to be written may exceed the memory capacity of the CPU module. In this case, perform any of the following operations.

- Delete the device comments of unused devices.
- Set the write target of device comments to SD memory card.
  For FX5CPU, the writable capacity of SD memory card is the same as that of CPU module.
- Set the range to be written.

Reading data from programmable controller

Read data from a CPU module.

For the considerations for reading data, refer to the following section.

☞ Page 256 Considerations for Online data operation

Operating procedure

1. Select the [Read] tab on the "Online Data Operation" screen.
2. Select the files to be read and read target.
3. Click the [Detail] button to set the details such as a range to be read.
4. Click the [Execute] button.

Point

Data can be read when a project is not opened.

☞ Page 62 Creating new project by data read from programmable controller

Precautions

When accessing SD memory card frequently using the data logging function or database function, the reading operation from the programmable controller (to open the "Online Data Operation" screen) may be slow.

Parameters

■Unsupported modules

The module parameters that are not supported by GX Works3 cannot be read.

■Model name in Safety Communication Setting

The model name of "Safety Communication Setting" of the module parameter is not written to CPU module. The item will be blank if it is read.

Program

When reading a program (local labels, program), read "Program".

■Initial values of local labels

The operation differs between writing to and reading from the programmable controller.

FX5CPU does not support it.

- When writing data to a programmable controller; write the local labels as "Local Label Initial Value".
- When reading data from a programmable controller; read the program to read "Initial Value" of local labels

■Execution type which is overwritten at reading program

- When reading CPU parameters and program file at once: The program file is overwritten with the execution type in accordance with the program setting of the read CPU parameter.
- When reading only a program file: The program file s overwrite in accordance with the parameter settings in GX Works3.
- When the CPU parameter setting does not exist in both CPU module and GX Works3: The program file is overwritten with "No Execution Type".

■Setting read range

Set the range of a program (program file) to be read from a CPU module on the "Program Detail Setting" screen displayed by clicking the [Detailed] button.

In addition, the secured steps for online program change can be obtained from CPU module.

FX5CPU does not support it.

Global label

■Initial value, accessing from external device

The operation differs between writing to and reading from the programmable controller.

FX5CPU does not support the initial value of labels nor the setting for accessing from external device.

- When writing data; global labels are written as "Local Label Initial Value" and "Global Label Assignment Information".
- When reading data; the information of "Initial Value" and "Access from External Device" of local labels are read with global labels by reading programs.

Restriction

When reading only a program in which global labels are used, the label definitions used in the program may not exist in the project. This may cause the disappearance of the labels on the ladder editor.
In this case, read the global label and program together.

Device memory

The device memory, which have the same name as the program, will be read as a file for a local device.

The device memory, which have the different name as the program, will be read as a file for a global device.

Only one device memory for a global device can be selected.

FX5CPU does not support local devices.

The device memory is read as a file for a global device regardless of whether or not it is consistent with the program name.

■Setting read target devices and their read ranges

Set the type and range of device memory to be read from a CPU module on the "Device Data Detail Setting" screen displayed by clicking the [Detailed] button.

Device memory can be read from link memory or buffer memory.

The devices that can be read are as follows:

○: Readable, —: No corresponding device

Device	Global device	Local device*1
M, V*1, T, ST, C, D, R*2, SA\M, SA\T, SA\ST, SA\C, SA\D	○	○
X, Y, L, B, F, SB, S, LT*1, LST*1, LC, W, SD, SW, SM, Z, LZ, RD*1, SA\X, SA\Y, SA\B, SA\W, SA\SD, SA\SM	○	—

*1 Not supported by FX5CPU.

*2 Not supported by RCPU.

Device comment

■Setting read range

Set the range of Common device comments/Each program device comments to be read from a programmable controller on the "Device Comment Detail Setting" screen displayed by clicking the [Detail] button.

Device initial values

Even when the device initial value read from the CPU module is local device, it is not displayed as a local device (# is not added) on GX Works3.

Deleting data in CPU module

Delete the data such as programs and parameters in the CPU module.
For a safety project, safety data can be deleted when the safety operation mode of RnSFCPU is in test mode.

Operating procedure

Select the file to be deleted from the list on the "Online Data Operation" screen, and click the [Execute] button.

Considerations for Online data operation

Unlocking a remote password

The confirmation message for unlocking the remote password is displayed when the remote password is set to the CPU module to be accessed. Unlock the password by following the message.
For details, refer to the following section.
☞ Page 310 Restricting Access from Other Than Specific Communication Route

13.2 Verifying Programmable Controller Data

This section explains how to verify the open project against the data in the CPU module.

The Verify with PLC function is used to compare the content of two projects or to locate the changes made in programs.

The projects set the security function can be verified when both the verify source data and the verify destination data are not read-protected.

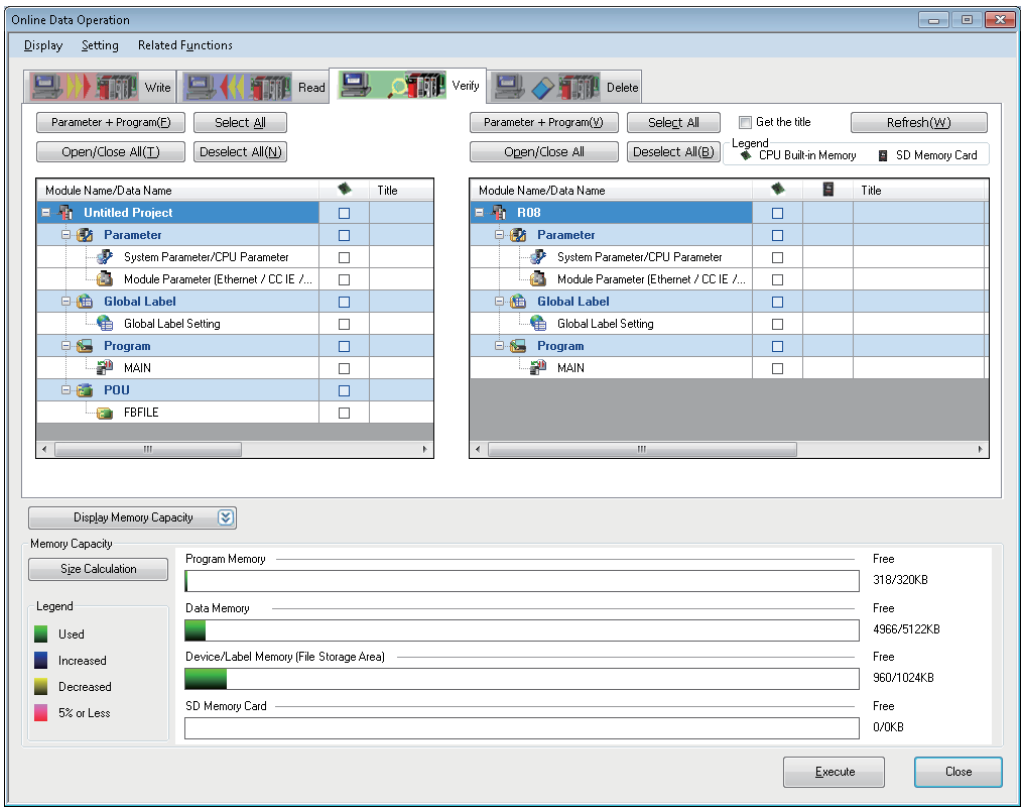
To verify data between two projects, use the Project Verify function.

For details, refer to the following section.

☞ Page 77 Verifying Projects

Window

[Online] ⇨ [Verify with PLC]



Operating procedure

Select the data to be verified, and click the [Execute] button.

The operation method for displaying the verification result is the same as the Project Verify function. Refer to the following section.

☞ Page 78 Checking verification result

Parameter

■Verification of module parameters

The data which can be selected as verification target are as follows: Ethernet modules, CC-Link modules, CC-Link IE Controller Network modules, CC-Link IE Field Network modules, and MELSECNET/H network modules.

13.3 Writing Programs While CPU is in RUN

This section explains the operations for changing programs or data while the CPU module is in RUN (online program change).
Before performing the online program change function, be sure to fully understand the descriptions below.
The specifications for online program change of CPU module, refer to the following manual.
📖 MELSEC iQ-R CPU Module User's Manual (Application)
There are two ways to write data while CPU module is in RUN as shown below:
○: Writable, ×: Not writable

Type	Description	Supporting CPU module		
		RnCPU/RnENCPU/ RnPCPU	RnSFCPU	FX5CPU
Online program change*1	Changes programs or data partially while the CPU module is in RUN	○	○*2	○
File batch online change	Writes data in file units while the CPU module is in RUN	○	○*2	×

*1 SFC program supports the online program change function only performed from the Zoom.
*2 Only standard data is writable.

Writing data while the CPU module is in RUN affects scan time due to the change of number of steps. Set the secured steps for online program change on the "Program Detail Setting" screen displayed by clicking the [Detailed] button of "Program".

Considerations

- The program execution is suspended while performing the online program change function. The length of down time depends on the number of the changed steps.
- When writing programs which includes the rise instructions, fall instructions, SCJ instructions, and STMR instructions while the CPU module is in RUN, the system may not operate properly.
- Do not perform the online program change function to the same program from multiple GX Works3 at the same time.
- If failed to write data while the CPU module is in RUN state, the project will be returned to the previous status before the conversion in order to make it possible to perform the online program change function again.
- Make sure that the parameters between in the CPU module and in the project mach before performing the online program change function.
- Make sure that the writing target file has been written to the CPU module before performing the online program change function.
- When "Check at Communication Time" has been set less than 90 seconds, the timeout will be checked in 90 seconds. If an error occurred, extend the timeout time on the "Specify Connection Destination" screen.
- The online program change function will continue performing if a warning occurred at conversion.
- When the online program change function is performed after changing the label names which can be accessed from external devices, the data before the change is remained in the global label assignment information. To delete the remaining data, write the global label assignment information from the "Online Data Operation" screen.
- When global labels are changed, programs using the global labels will be a writing target for performing the online program change function.

Design instructions

⚠WARNING

■When data change, program change, or status control is performed from a personal computer to a running programmable controller, create an interlock circuit outside the programmable controller to ensure that the whole system always operates safely.
Furthermore, for the online operations performed from a personal computer to a CPU module, the corrective actions against a communication error due to such as a cable connection fault should be predetermined as a system.

Startup/Maintenance Instructions

⚠CAUTION

■The online operations performed from a personal computer to a running CPU module (program change while a CPU module is in RUN state, operating status change such as RUN-STOP switching, and remote control operation) have to be executed after the manual has been carefully read and the safety has been ensured.
When changing a program while the CPU module is in RUN, it may cause a program corruption in some operating conditions. Fully understand the precautions described in this section.

Writing data after changing programs partially while CPU module is in RUN (Online program change)

This function partially overwrites data such as program files in the CPU module. Before changing the program, check if the programs between in the CPU module and in the project match using the "Verify with PLC" function. If there is no target file in the CPU module, the online program change function cannot be performed.
The list of operations for the data which can perform the online program change function and the target files are as shown below.
○: Applicable, —: Not applicable

Operation		Online program change target file				
		Program	POU	Local label initial value*1	Global label (settings, assignment information*1)	Global label initial value*1
Global label	Adding/changing global labels	—	—	—	○*2	○*3
Program block	Adding/changing local labels	○	—	○*3	—	—
	Adding/changing/deleting programs	○	—	—	—	—
Function block	Adding/changing local labels	—	○	—	—	—
	Adding/changing/deleting programs	—	○	—	—	—
Function	Adding/changing/deleting local labels	—	○	—	—	—
	Adding/changing/deleting programs	—	○	—	—	—

*1 Not supported by FX5CPU.
*2 The operation for the assignment information can be performed only when the labels are set to be accessed from external devices.
*3 The operation for the label initial value file can be performed only when initial values are set to labels.

Operating procedure

1. Verify a program against a program on the programmable controller to ensure that they match.
If the program part to be written does not match with the program on the CPU module, the online program change function cannot be performed. Perform the function in file units. (☞ Page 261 Writing data in units of files while the CPU module is in RUN (file batch online change))

2. Modify a program.

3. Select [Convert] ⇨ [Online Program Change].

4. Select the target program, and click the [OK] button.

Point

Check the processing procedure and precautions for the online program change function by clicking the [Notes] button.

Online program change after editing device comments

When the online program change function is performed after editing device comments, the device comments are written after writing programs.

The operations below can be chosen for writing device comments by selecting [Tool] ⇨ [Options] ⇨ "Convert" ⇨ "Online Program Change".

Item	Value	Description
Check Device Comment Difference	No	The confirmation message does not appear, and all device comments are written.
	Yes	The confirmation message appears, and whether or not to write device comments can be selected.

Note that the online program change function cannot be performed even if only device comments were edited with the program or labels unchanged.

When writing only device comments while the CPU module is in RUN, write data to the programmable controller from the "Online Data Operation" screen.

Online program change of global label/local label initial values (Version 1.000A only)

After adding/modifying global label initial values or local label initial values, writing the label initial value file to CPU module is required.

FX5CPU does not support the initial value of label.

When clearing all initial values, deleting the label initial value file to CPU module is required.

Furthermore, under booting, writing the label initial value file to SD memory card is required and deleting the label initial value file is also required from the Boot File Setting of the Memory Card Parameter when clearing all initial values. If do not write/delete the label initial value file to/from the CPU module, an error occurs at powering ON, resetting, or changing the status from RUN to STOP of the CPU module.

Online program change with the smallest configuration

The data with the smallest configuration can be written by setting the option as shown below. Set the option for saving the data capacity.

Note that, however, when the project with the smallest configuration is read from the programmable controller, the project cannot be restored completely. Always save the recent write source project.

- [Tool] ⇨ [Options] ⇨ "Convert" ⇨ "Online Program Change" ⇨ "Operational Setting"

Writing range for online program change

The range to be written differs depending on the editing methods of programs.

When a new ladder block is inserted, or a ladder block is deleted, the inserted program is written together with one instruction after the inserted ladder block to the CPU module.

Therefore, the online program change function may not be completed at a time depending on the number of program steps before and after addition or deletion. In this case, reduce the number of steps written at a time and perform the function in several times.

Execution of rising/falling instructions after performing online program change

When the raising/falling instructions in the subroutine FB program were modified or added, the instructions are not executed immediately after performing the online program change function even if the execution condition was satisfied.

The instructions are executed next time the execution condition is satisfied.

<Example of project configuration>
- MAIN1 : Program file
  - FbPou1 : Subroutine FB program (Modified/added)
- MAIN2 : Program file
  - FbPou2 : Subroutine FB program (Modified/added)
    - FbPou3 : Macro FB program (Modified/added)*1

*1 The macro FB program used in the subroutine FB program is the same as subroutine FB program.

Writing data in units of files while the CPU module is in RUN (file batch online change)

Write programs or data in units of files while the CPU module is in RUN state.
Data such as sequence programs (standard programs) and device comments can be written by performing the file batch online change function.
FX5CPU does not support it.

Operating procedure

Select [Online] ⇨ [Write to PLC].

Precautions

- Writing data cannot be performed to the program files which are not registered in the parameter setting.
- The sequence program in which the global labels or FBs/FUNs are used and the global label setting file or FB files that are referred from the sequence program cannot be written simultaneously in file units while CPU module is in RUN. Change the CPU module status to STOP/PAUSE before writing the global label setting file or FB files.

13.4 Reading/Writing/Deleting User Data

Read/write/delete user data from/to the CPU built-in memory/SD memory card.

FX5CPU does not support it.

The files which have a three-character extension (example: csv, txt, bin, xml) can be used as a user data.

Note that, the following files are excluded.

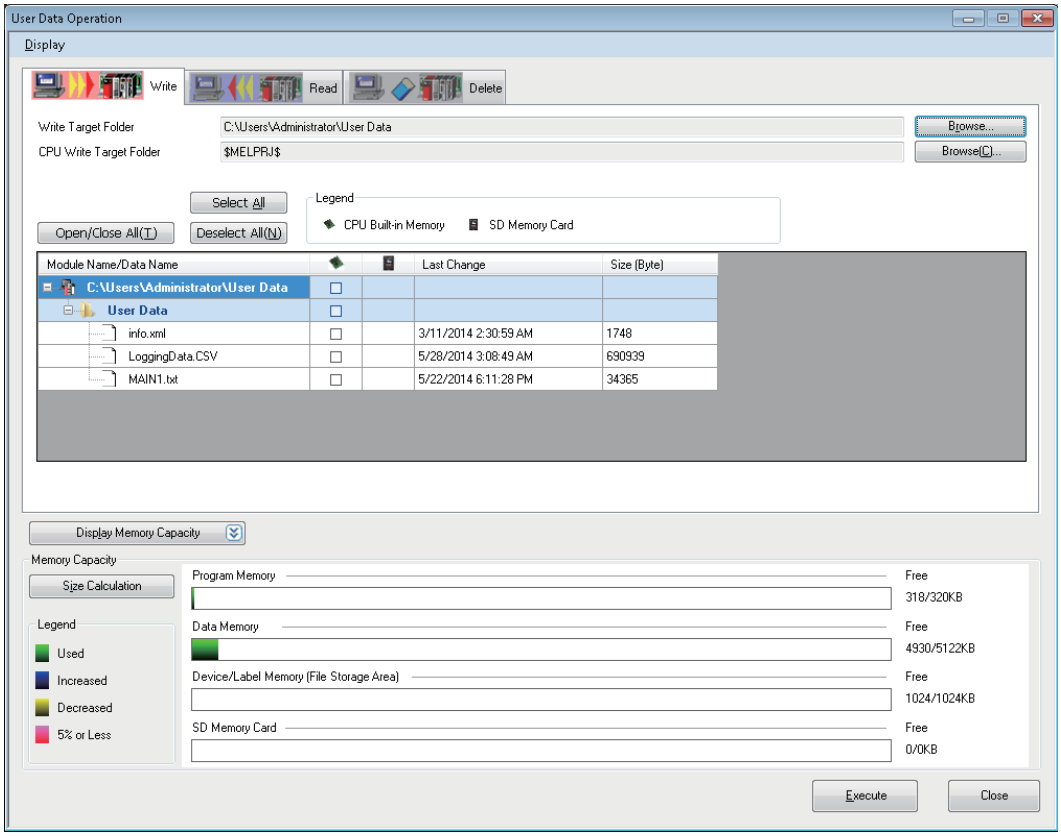
- SYS, PRG, PFB, PRM, DID, DCM, QDR, EVENT.LOG (includes file name), QST, IFG, LID, CAB, LCS, LIS, SYP, a file that '$' is attached in front of the file name

The user data written to the CPU built-in memory/SD memory card can be used in sequence programs.

Window

[Online] ⇨ [User Data] ⇨ [Write]/[Read]/[Delete]

The following screen is an example when writing data.



Operating procedure

1. Select [Write], [Read], or [Delete] tab on the "User Data Operation" screen.
2. Select the target files, and a write target or read source.
3. Click the [Execute] button.

Creating/deleting folders, changing folder name

Folders can be created in the CPU built-in memory or SD memory card.
The created folder name can be changed and deleted.
Create/delete a folder, or change a folder name on the "Browse Folders" screen displayed by clicking the [Browse] button on the "User Data Operation" screen.

■Creating folders

Select the place in which the folder is to be created on the "Browse Folders" screen, right-click it and select [Create Folder] from the shortcut menu.

■Deleting folders/changing folder name

Select the target folder on the "Browse Folders" screen, right-click it and select [Delete Folder]/[Change Folder Name] from the shortcut menu.

13.5 Deleting All Files (Initialization)

Perform initialization to delete files in a CPU module or a memory card.
This function is used to use the CPU module and the memory card for the first time, or to clear all data stored in them.
Once initialized, the following processing is performed to each memory:

- Data memory: Delete all folder/all files in the program memory and data memory.
- Device/label memory: Delete all files in the file storage area in the device/label memory.
- SD memory card: Delete all folders/all files in the SD memory card.

Operating procedure

Delete all files on the "CPU Memory Operation" screen. Refer to the following section.
☞ Page 331 Initializing/Clearing Memory

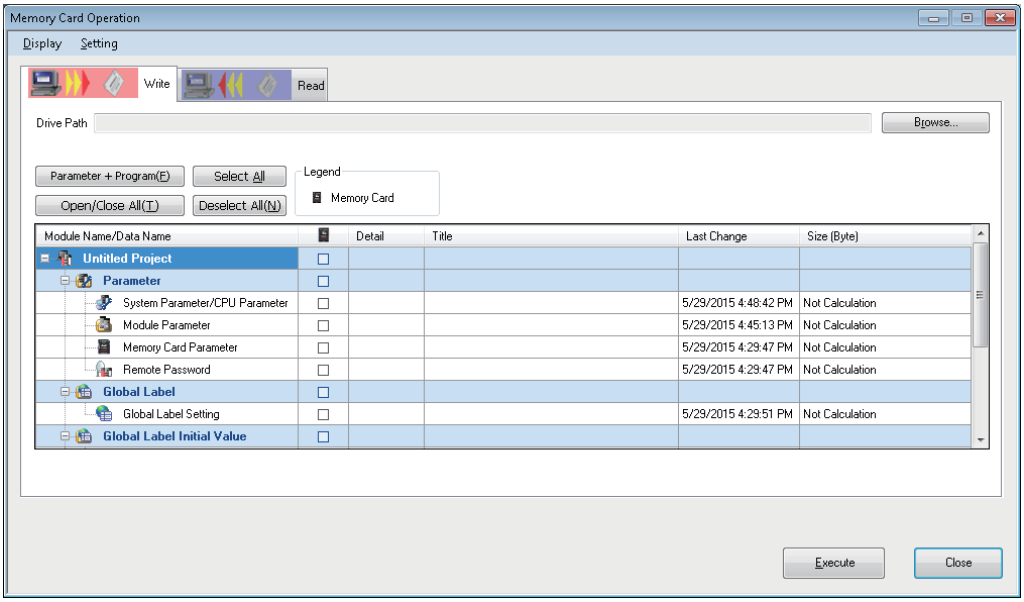
13.6 Writing/Reading Data to/from Memory Card

This section explains how to writing/reading data to/from a memory card attached to a personal computer.
This operation is used, for example, when data can not be written/read by attaching a memory card to a CPU module.
However, this writing/ reading function cannot be used in a safety project.
For details on the writing/reading data to a CPU module attached a memory card, refer to the following section.
☞ Page 248 Writing/Reading Programmable Controller Data

Configuration of Memory Card Operation screen

Window

[Tool] ⇨ [Memory Card] ⇨ [Write to Memory Card]/[Read from Memory Card]
The following screen is an example when writing data.



The data name in gray indicates that it is in the unconverted state.

Writing to/Reading from memory card

Write/Read data to/from a memory card.
The file which is in the '$MELPRJ$' folder of the specified drive path is the target to be read/written.

Operating procedure

1. Select the [Write] tab/[Read] tab on the "Memory Card Operation" screen.
2. Set the drive path of the write target/read source, then click the [Execute] button.

14 CHECKING OPERATION OF PROGRAMS

This chapter explains how to check the execution status of Intelligent function modules and CPU modules connected to a personal computer.

The functions to check the execution status are as follows:

Purpose	Function name	Reference
A function to check the status of the programs being executed on the program editor.	Monitor	Page 268 Check Execution Programs on Program Editor
A function to check the current values of the device or buffer memory in batch.	Device/Buffer Memory Batch Monitor	Page 275 Checking Device/Buffer Memory in Batch
A function to check the current values by registering devices and labels.	Watch	Page 277 Checking Current Values by Registering Devices/Labels
A function to check the processing time of programs being executed.	Program List Monitor*1	Page 280 Checking Processing Time of Program
A function to check the number of executions of interrupt programs used in a program.	Interrupt Program List Monitor*1	Page 281 Checking Execution Counts of Interrupt Programs
A function to check the I/O signals and the current value of buffer memory by registering the module information of Intelligent function module.	Intelligent Function Module Monitor*1	Page 282 Checking Current Values in Intelligent Function Module
A function to save and check the device values of a CPU module at any timing to a personal computer.	Memory Dump*1	Page 284 Saving and Checking Device Value
A function to check the sampled data (memory dump result file) on the program editor.	Offline Monitor	Page 288 Checking Sampled Data on Program Editor

*1 Not supported by FX5CPU.

Changing current values

Check the operation by changing the current value of devices, labels, and buffer memory using the Change Current Value function.

This function turns bit devices ON/OFF forcibly on the CPU module. The current values of the word device and buffer memory can be changed forcibly as well.

The current values can be changed on the following screens.

- Program editor
- "Device/Buffer Memory Batch Monitor" screen
- Watch window
- Intelligent Function Module Monitor window

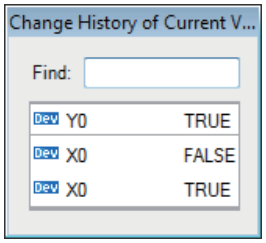
Safety devices/labels and standard/safety shared global labels can be changed when the safety operation mode of RnSFCPU is in the test mode.

■Change history of current value

When the current values of bit device are changed, the change history can be checked.

Window

[Debug] ⇨ [Change History of Current Value]



14.1 Monitor Status

Check the operation status of the CPU module and Intelligent function modules using the monitoring function by connecting a personal computer.

Starting/stopping monitoring

Start/stop monitoring by selecting one of the following menus.

- [Online] ⇨ [Monitor] ⇨ [Start Monitoring]( )/[Stop Monitoring]( )
- [Online] ⇨ [Monitor] ⇨ [Start Monitoring (All Windows)]( )/[Stop Monitoring (All Windows)]( )
- [Online] ⇨ [Watch] ⇨ [Start Watching]/[Stop Watching]
- Select [Online] ⇨ [Monitor] ⇨ [Monitor Mode] on the active ladder editor

For a ladder editor, start monitoring by selecting a mode from the pull-down menu on the mode display area located at the upper left of the ladder editor.

■When monitoring multiple projects using a single personal computer

- If an communication error has occurred in one of the projects, the monitoring speed of the other projects may become slow. The operation speed will be back to normal after closing the error message displayed in the project in which the communication error occurred.
- The following functions may not be performed normally:
  Program List Monitor and Interrupt Program List Monitor

■Monitoring buffer memory and link devices

Monitor the ON/OFF state of the buffer memory or link devices (example: U0\G0.1) by setting the following option.

- [Tool] ⇨ [Options] ⇨ "Monitor" ⇨ "Common Item"/"(Program editor)" ⇨ "Operational Setting"

■Changing display format of word devices

Select [Online] ⇨ [Monitor] ⇨ [Change Value Format (Decimal)] / [Change Value Format (Hexadecimal)] to change the display format of the monitored values.

■FB instance

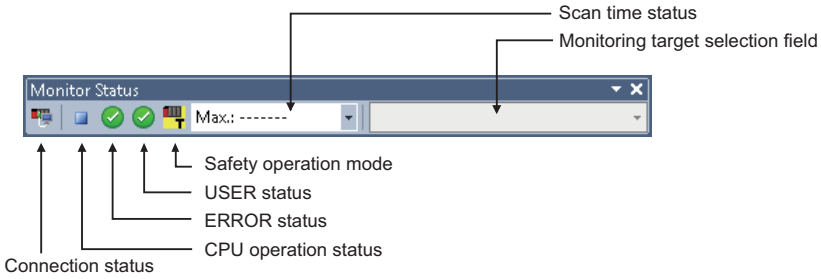
When monitoring the devices/labels in an FB instance, open the FB program and select the FB instance to be monitored from the monitor status bar.

Monitoring status

Monitoring status is displayed during monitoring on the work window.

Window

At the start of monitoring



Displayed items

Item	Description	Icon	Detail
Connection status	Displays the connection status between a CPU module and personal computer. The "Specify Connection Destination" screen is displayed by clicking the icon. For more details, refer to the following section. ☞ Page 235 SETTING ROUTE TO CPU MODULE		When connected to a CPU module
CPU operation status	Displays the CPU module status operated by the key switch on the CPU module or the remote operation from GX Works3. The "Remote Operation" screen is displayed by clicking the icon. For more details, refer to the following section. ☞ Page 326 CPU Module Remote Operation		RUN
			STOP
			PAUSE
ERROR state	Displays the ERROR LED state of CPU module. The "Module Diagnostics" screen is displayed by clicking the icon. For more details, refer to the following section. ☞ Page 314 Diagnosing CPU Module		ERROR is OFF.
			ERROR is ON.
			ERROR is flashing.
USER state	Displays the USER.LED state of CPU module. The "Module Diagnostics" screen is displayed by clicking this icon. For more details, refer to the following section. ☞ Page 314 Diagnosing CPU Module		USER is OFF.
			USER is ON.
			USER is flashing.
Safety operation mode	Displays the safety operation mode. The "Switch Safety Operation Mode" screen is displayed by clicking the icon. For details, refer to the following section. ☞ Page 327 Switch Safety Operation Mode		Safety mode
			Test mode
Scan time state	Displays the current value, maximum value, and minimum value of the scan time by switching them from the pull-down list.		
Monitor target selection	When monitoring multiple FB programs, select the target FB instance to be monitored.		

Monitor mode

Ladder editors are switched to monitor mode during monitoring.

☞ Page 123 Configuration of ladder editor

14.2 Check Execution Programs on Program Editor

Use the monitoring function to check the following execution programs on each program editor.

- Ladder program
- ST program
- FBD/LD program
- SFC program

Open a program editor to be monitored in advance.

Point

The display format and detailed operations for each function can be set by the following option setting.
[Tool] ⇨ [Options] ⇨ "Monitor"

Considerations when monitoring file registers

When monitoring file registers while "Use File Register of Each Program" is selected in "File Setting" of the CPU Parameter, the monitor operation will differ depending on the setting of Device/Label Access Service Processing Setting.
Therefore, add the programs to transfer the file registers to be monitored to the devices before monitoring the transferred programs. Note that the number of steps and scan time will be lengthened for the added programs.
If the file register set to "Use File Register of Each Program" is used as a data with no latch (such that the data is set to clear to '0' at the first time), replacing the file registers to local devices avoids increase of number of steps and scan time due to the addition of the programs.

- When "Execute END Processing between Programs" is selected:
  When monitor is requested after the program which is set not to use the file register, FFFFH (-1) will be monitored since the monitor is performed between programs or at the END process.
- When other than "Execute END Processing between Programs" is selected:
  The value of the file register file that is enabled in the program executed at the right before the END processing is monitored.
  Example) When the execution order of the program is 'A→B→C→(END processing)→A→B', monitor the value of the file register at execution of program C.

Monitoring possibility for devices

The following table shows the possibility to monitor devices on a program editor.

Possibility	Device
Possible	X, Y, M, L, B, F, SB, V, S*1, T*2, T(TS), T(TC), T(TN), ST*2, ST(STS), ST(STC), ST(STN), LT*2, LT(LTS), LT(LTC), LT(LTN), LST*2, LST(LSTS), LST(LSTC), LST(LSTN), C*2, C(CS), C(CC), C(CN), LC*2, LC(LCS), LC(LCC), LC(LCN), D, W, SW, FX, FY, SM, SD, J□\X, J□\Y, J□\B, J□\W, J□\SW, U□\G, U3E□\G, U3E□\HG, Z, LZ, R, ZR, RD, BL, BL□\S
Not possible	FD, N, P, I, J, U

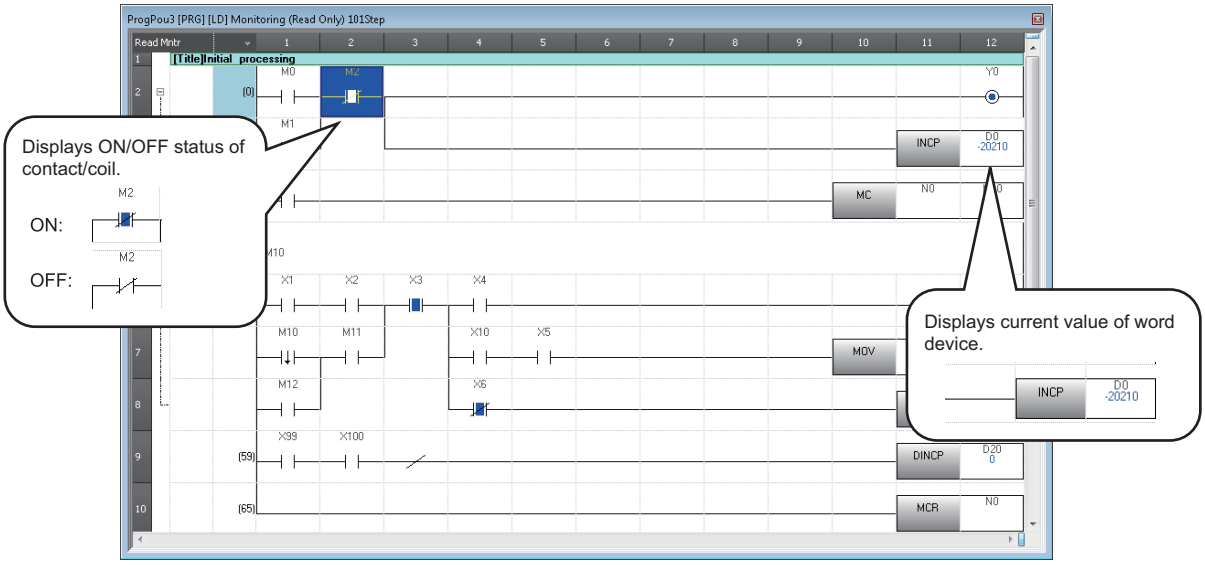
*1 'S' can be monitored in a Zoom or on an SFC diagram only. Monitor BL□\S when using program editors not a Zoom.
*2 Same value as 'N' (current value) is displayed.

Ladder

Circuit monitor

Operating procedure

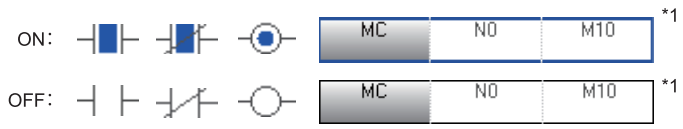
Open the ladder editor, and select [Online] ⇨ [Monitor] ⇨ [Start Monitoring] ( )/[Stop Monitoring] ( ).



When any values other than fixed value are specified to the element of an array, an indefinite value is displayed as the monitored value of the array.

■ON/OFF state display

During monitoring, the ON/OFF state is displayed as shown in the following figure.



*1 Only comparison instructions that are equivalent to contacts and the instructions that are equivalent to coils are supported.
Comparison instructions equivalent to coil: BIN16-bit data comparison, BIN32-bit data comparison, floating-point data comparison, 64-bit float data comparison
Instructions equivalent to coil: SET, RST, PLS, PLF, SFT, SFTP, MC, FF, DELTA, DELTAP

■Monitoring FB programs

Double-click the FB instance to monitor the FB program.

Changing current values

Current values can be changed by the following methods during monitoring.

Target device/label	Operating procedure
Bit type	Select a cell of device/label, then press the Shift + Enter keys.
Word type	Register the device/label in the Watch window, and change the value. (☞ Page 277 Checking Current Values by Registering Devices/Labels)

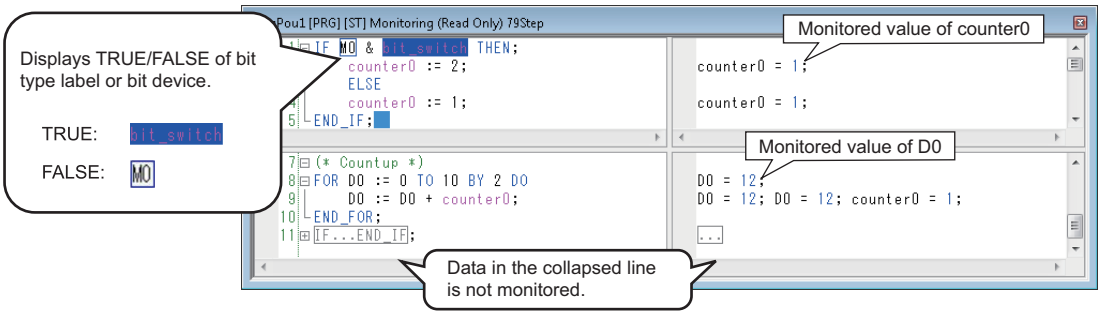
ST

The monitored values of bit type are displayed on a program, and the monitored values of labels other than bit type and word devices are displayed on the right side of the split window.
The monitored value is displayed on the tooltip by placing the cursor on a device/label name.

Monitor

Operating procedure

Open the ST editor, and select [Online] ⇨ [Monitor] ⇨ [Start Monitoring] ( )/[Stop Monitoring] ( ).


Displays TRUE/FALSE of bit type label or bit device.
TRUE:
bit_switch
FALSE:
M0
Pou1 [PRG] [ST] Monitoring (Read Only) 79Step
Monitored value of counter0
Monitored value of D0
Data in the collapsed line is not monitored.

Changing current values

Current values can be changed by the following methods during monitoring.

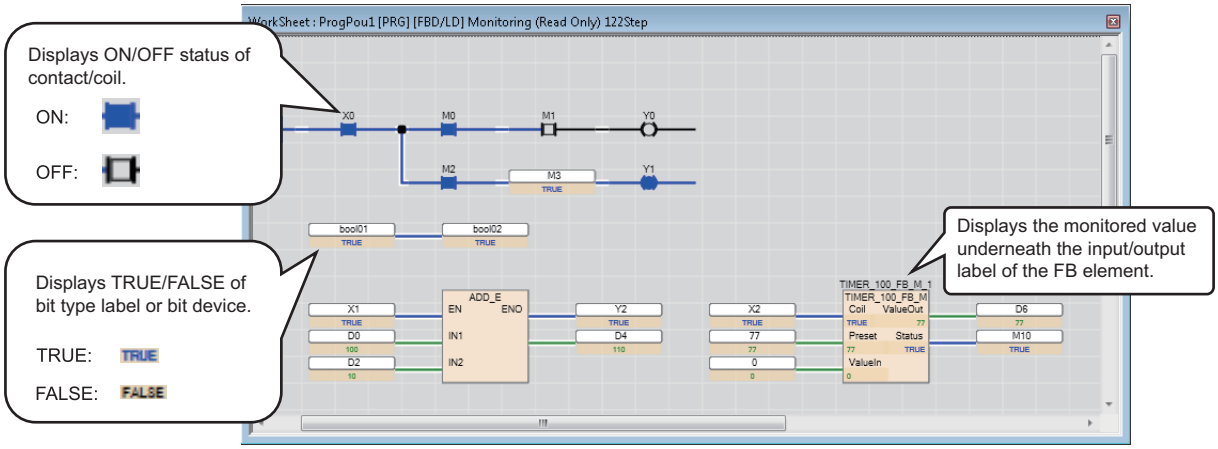
Target device/label	Operating procedure
Bit type	Select a token of device/label, then press the Shift + Enter keys.
Word type	Register the device/label in the Watch window, and change the value. (☞ Page 277 Checking Current Values by Registering Devices/Labels)

FBD/LD

Monitor

Operating procedure

Select [Online] ⇨ [Monitor] ⇨ [Start Monitoring] ( )/[Stop Monitoring] ( ).



When the current values cannot be monitored, "---" is displayed.

Precautions

When the type of the defined function block is macro type, monitoring the input label depends on the monitor condition of the connected element. Therefore, the input label also cannot monitored when the connected element cannot be monitored.

Changing current values

Current values can be changed by the following methods during monitoring.

Target device/label	Operating procedure
Bit type	Select an element of device/lavel, then press the Shift + Enter keys.
Word type	Register a device/label in the Watch window, and change the value. (☞ Page 277 Checking Current Values by Registering Devices/Labels)

SFC

Monitor device values of the SFC program that the execution type of the program is set for "Scan".

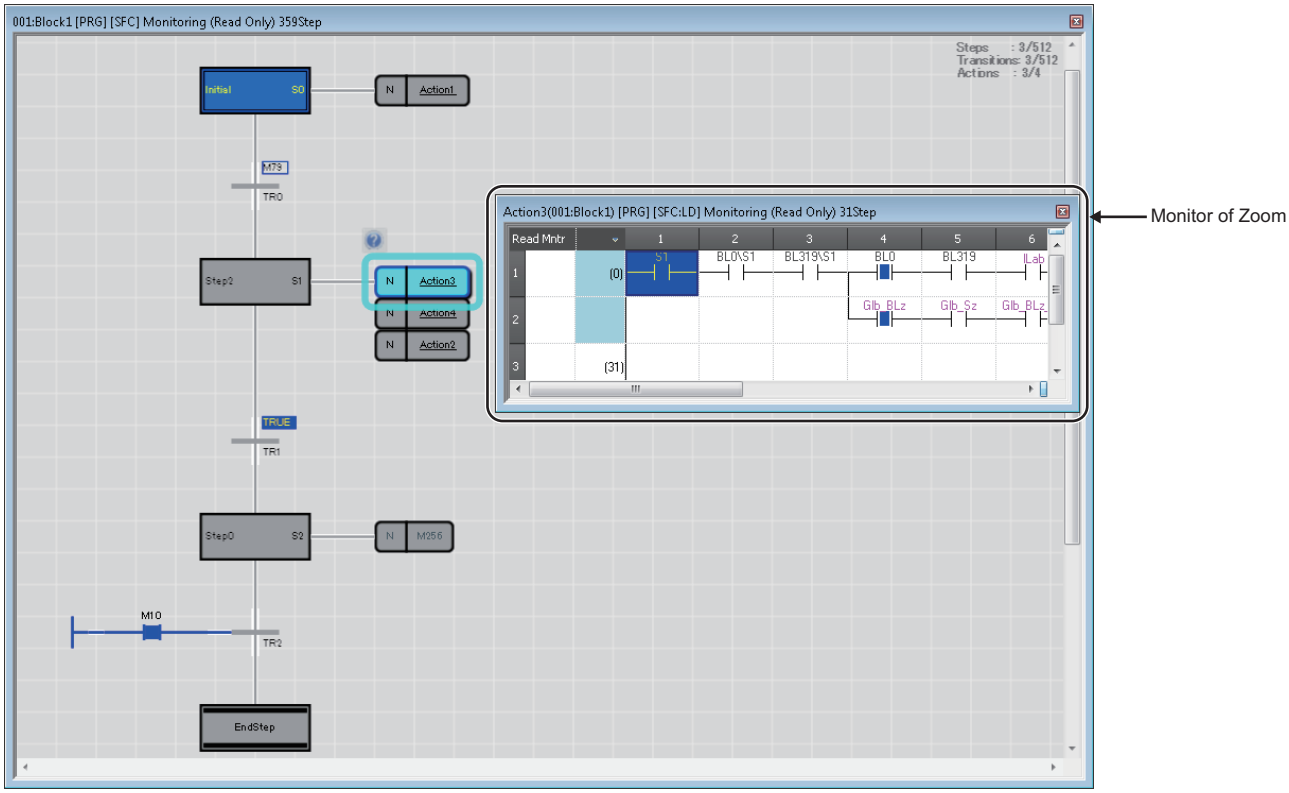
SFC program has the monitoring functions shown below.

Purpose	Target editor/function name	Reference
To check the active status of steps and current values of devices on an SFC diagram.	SFC Diagram	Page 272 Monitoring SFC Diagram
To check the current values of devices in a Zoom.	Zoom	Page 273 Monitoring Zoom
To check block information in a list.	SFC block list	Page 273 Monitoring SFC block list
To check active status of all blocks in a list.	SFC All Blocks Batch Monitor	Page 274 Monitoring all SFC blocks in batch/active steps
To check active status of steps in a specified block in a list.	Active Step Monitor	

Monitoring SFC Diagram

Operating procedure

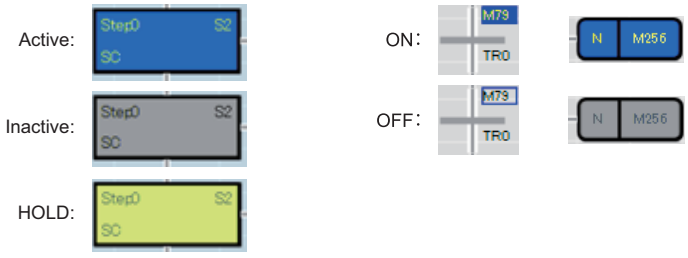
Select [Online] ⇨ [Monitor] ⇨ [Start Monitoring] ( )/[Stop Monitoring] ( ).



■SFC element status display

During monitoring, the SFC element status is displayed as shown in the following figure.

Monitor status of FBD/LD elements is same as the one of FBD/LD editor.



Monitoring Zoom

Select [View] ⇨ [Open Zoom/Start Destination Block], then open the Zoom to monitor.

The methods for operating/displaying a monitor of Zoom are same as the ones of each program editor.

Devices that can be monitored are same as the ones of each program editor except for S□. S□*1 can be monitored only in a Zoom.

*1 The devices may not be monitored properly when a target block is switched by the BRSET instruction.

Changing active status

Active status of blocks/steps can be changed by the following methods during monitoring.

It can be used to check the operation when starting/stopping only a particular block/step.

Target	Operation screen	Operation
Block	Each program editor (including a Zoom)	Change the current value (TRUE/FALSE) of BL□.*1
	Device/Buffer Memory Batch Monitor	
	Watch window	
Step	SFC diagram editor	Select a step, then select [Debug] ⇨ [Control SFC Step] ⇨ [Activate the Selected Step]/[Deactivate the Selected Step]/[Activate the Selected Step Only].
	Zoom	Change the current value (TRUE/FALSE) of S□ or BL□\S□.*1
	Each program editor (excluding a Zoom)	Change the current value (TRUE/FALSE) of BL□\S□. *1
	Device/Buffer Memory Batch Monitor	
	Watch window	

*1 Blocks/steps can be activated by changing the current values of labels which were assigned BL□\S□ or BL□.

Precautions

Note that the control of a CPU module is affected by activating/deactivating steps/blocks.

Changing current values (device/label)

Current values of the following devices/labels can be changed on an SFC diagram during monitoring. Select a device/label and press the Shift + Enter keys.

- Bit devices/bit type labels of transitions/actions on the SFC diagram editor
- Bit devices/bit type labels of FBD/LD elements connected to transition

The methods for changing current values of devices/labels in a Zoom is same as the one of each program editor.

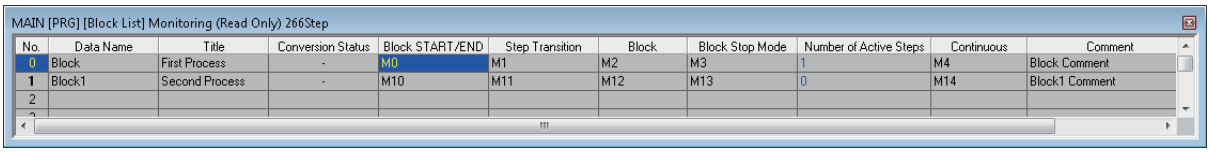
For S□, BL□\S□, BL□, refer to the following section.

☞ Page 273 Changing active status

Monitoring SFC block list

Display current block information on an SFC block list.

Select [View] ⇨ [Open SFC Block List], then open an SFC block list to monitor.


MAIN [PRG] [Block List] Monitoring (Read Only) 266Step
No. Data Name Title Conversion Status Block START/END Step Transition Block Block Stop Mode Number of Active Steps Continuous Comment
0 Block First Process - M0 M1 M2 M3 1 M4 Block Comment
1 Block1 Second Process - M10 M11 M12 M13 0 M14 Block1 Comment
2

Point

The SFC diagram of specified block can be displayed by double-clicking the block column during monitoring.

Monitoring all SFC blocks in batch/active steps

Active status of all blocks/steps can be displayed.

Active status of /blocks/steps are not changed if the status of a CPU module is switched from RUN to STOP. Thereby, the active status when the CPU module was changed to STOP state is displayed on a monitor screen.

Precautions

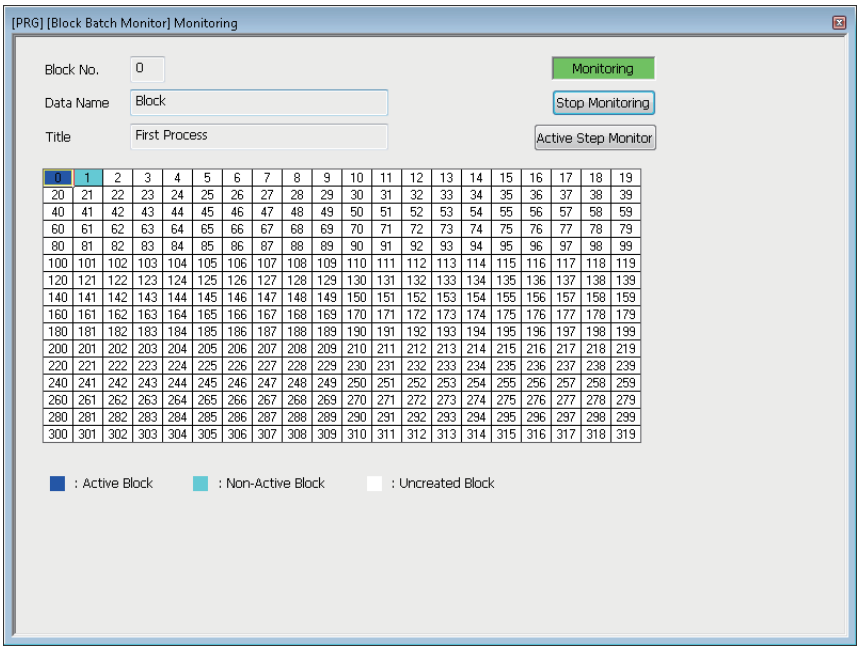
This monitoring function may not work properly when there is a difference between the program written in a CPU module and program in a project.

■Monitoring SFC blocks in batch

Display current active/inactive status of all blocks in a list.

Window

Select [Online] ⇨ [Monitor] ⇨ [SFC All Blocks Batch Monitoring] ( ).



■Monitoring active steps

Display current active/inactive status of all steps that exist in a specified block in a list.

Up to five monitor screens can be displayed at once.

Operating procedure

Move the cursor on a block to be monitored on the "Block Batch Monitor" screen, then click the [Active Step Monitor] button.

14.3 Checking Device/Buffer Memory in Batch

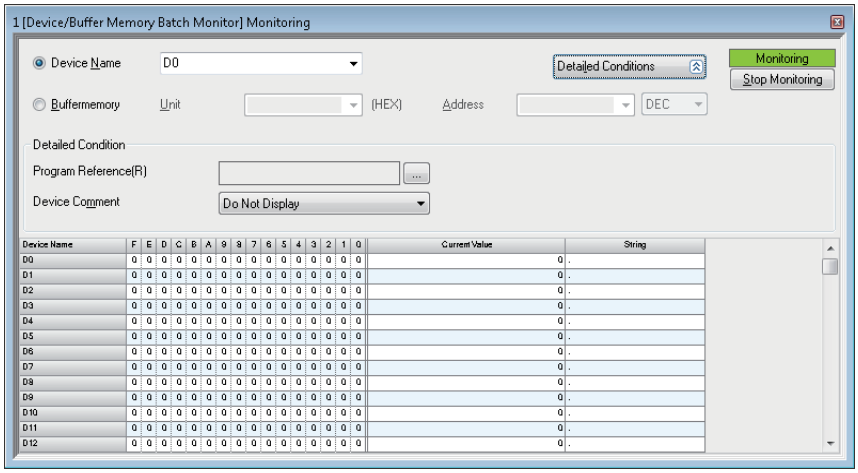
Check the devices and buffer memory using the Device/Buffer Memory Batch Monitor function.
When the multiple monitoring screens are opened, lead time to start monitoring and monitoring intervals may become longer.
Multiple CPU shared memory (buffer memory) of multiple CPU system can be monitored only by the Device/Buffer Memory Batch Monitor function.

Window

[Online] ⇨ [Monitor] ⇨ [Device/Buffer Memory Batch Monitor] ( )

●Toolbar

●Device/Buffer Memory Batch screen



Up to 64 monitoring screens can be displayed at once. The numbers are displayed at the start of the screen title.

Operating procedure

■Monitoring devices in batch

Enter the target (start) in "Device name".
For the devices that can be entered, refer to the following section.
☞ Page 337 Applicable Devices in GX Works3

- When the target device is local device: Specify "Program Reference".
- When checking the TC setting values: Enter a device of timer, retentive timer, long timer, long retentive timer, counter, or long counter (Example: T10, ST10, LT10, LST10, C10, LC10) and specify "Program Reference".

■Monitoring buffer memory in batch

Enter the start I/O number and address of the Intelligent function module.

Item	Description
Unit	Enter the start I/O number of the Intelligent function module to be monitored in hexadecimal format. • For multiple CPU configuration Specify the CPU module to be monitored. CPU No.1: 3E00, CPU No.2: 3E10, CPU No.3: 3E20, CPU No.4: 3E30
Address	Enter the address of buffer memory to be monitored in decimal/hexadecimal format.

■Monitoring blocks/steps in an SFC program in batch

Enter BL☐ in "Device name" when monitoring blocks, and enter BL☐\☐ there when monitoring steps.

Point

Fonts can be changed.
☞ Page 48 Checking and Changing Colors and Fonts

Changing current values

The current value can be changed by double-clicking the cell of the device or pressing the Enter during monitoring.
For word devices, change the current values by registering the target device to be changed into the Watch window. (☞ Page 277 Checking Current Values by Registering Devices/Labels)

- For TC devices, only contacts and current values can be changed.
- Since the digit specification (Example: K4DX0) cannot be used for the following devices, the current values can be changed only when "Bit and Word" or bit order of '0-F' is specified in Display Format ( ).
  DX, DY, FX, FY, J□\X□, J□\Y□, J□\B□, J□\SB□\
- Since the bit specification (example: Z0.0) cannot be used for Z device, the current values cannot be changed when "Bit and Word" in "Display format" or bit order of '0-F' is specified in Display Format ( ).
- For S□, BL□\S□, BL□, blocks/steps are activated/deactivated by changing the current values. (☞ Page 273 Changing active status)

Considerations when monitoring file registers

The operation of file registers differ depending on the settings of the CPU parameters. Refer to the considerations in the following section.
☞ Page 268 Considerations when monitoring file registers

Specifying reference target of program

Specify the monitoring target program when monitoring the setting values of timer devices and counter devices, or local devices.

Operating procedure

1. Click the [Detailed Conditions] and click [...] in "Program Reference" on the "Device/Buffer Memory Batch Monitor" screen.
2. Set the items on the "Program Reference" screen, and click the [OK] button.

14.4 Checking Current Values by Registering Devices/Labels

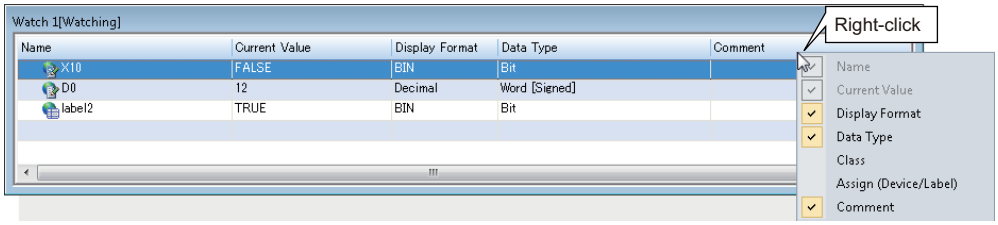
Check the current values by registering devices and labels using the Watch function. Register the target devices/labels on the Watch window.

Multiple devices/labels can be registered in batch by specifying a range.

Once devices/labels are registered and the project is saved, the registered devices/labels are displayed in registered status when the project is opened again.

Window

[View] ⇨ [Docking Window] ⇨ [Watch 1] to [Watch 4] ( )



The display format can be selected per row from the pull-down list displayed when selecting the display format.

Operating procedure

1. Register devices/labels to be monitored. (☞ Page 277 Register to Watch window)
2. Select [Online] ⇨ [Watch] ⇨ [Start Watching].

"[Watching]" is added on the Watch window title during monitoring.

Register to Watch window

Register devices/labels to be monitored on the Watch window.

Operating procedure

■Enter to register

1. Display the watch window.
2. Enter a device/label to be registered to "Name", and press the Enter key.

The input format of devices/labels are as follows:

- Global device: Device name
- Local device: Program name/#Device name ('#' is not necessary for index register (Z))
- Global device: Label name
- Local device: Program block name/label name
- Step relay: BL□\S□

When registering structures, function blocks, or arrays, specify them as follows:

Type	Target	Specification method
Structure	Label (Structure)	Label name
	Structure element	Label name.Element name
Function block	Label (Function block)	Label name
	Label in the function block	Label name.Label name in the function block
Array	Label (Array)	Label name excluding array notation
	Array element and the specific dimension part of array (more than two-dimensional)	Label name[one-dimensional element number][two-dimensional element number][three-dimensional element number]

Type	Target	Specification method
Structure array	Label (Structure array)	Label name
	Specific member	Label name[one-dimensional element number][two-dimensional element number][three-dimensional element number].member name
Nested function block	Label (Nested function block)	Label name
	Labels in the nested function block	Label name. label name in the function block. label name in the nested function block When nesting a label for one or more level deeper, the lower part of the label name which are delimited by a dot '.' is regarded as a label name. (Example: MAIN_PRG_LD/FB0_1.FB1_1.FB2_1.INOUT)
	When a function block includes array labels or structured labels	Follow the specification method of the array or structure.

■Registering devices/labels from program editor/label editor

1. Select a device/label to be registered in the Watch window from each program editor/label editor.
2. Select [Online] ⇨ [Watch] ⇨ [Register to Watch Window] ⇨ [Watch 1] to [Watch 4].

Devices/labels can be registered by dragging and dropping onto the Watch window.

- Ladder editor*1: Select a cell of device/label to be registered, and then drag and drop the border of the selected cell onto the Watch window.
- ST editor*1: Select a token of device/label to be registered, and then drag and drop it onto the Watch window.
- FBD/LD editor*1: Select an element of device/label to be registered, and then drag and drop it onto the Watch window with pressing Ctrl key.
- SFC diagram editor: Select a step to be registered, or a transition/action that set device/label, and then drag and drop it onto the Watch window.
- Label editor: Select the header of a label to be registered, and then drag and drop it onto the Watch window

*1 For RCPU, step relays (S☐) cannot be registered except in a Zoom.
Step relays (S☐) are registered as step relays with a block specification (BL☐\S☐) automatically in a Zoom.

Point

Rectangular selection can be made by dragging the devices/labels while pressing the Alt key on the ST editor. The devices/labels only in the selected range can be registered.

Automatic registration to Watch window

Automatically register the devices/labels in the specified range on the program editor to the Watch window.
The registration target for watch window is updated each time when a specified range is changed.
The following table shows the range of automatic registration for each program language.

Program language	Range	Remarks
Ladder	Device/label on the ladder block which contains the selected cell	The devices/labels in FB and inline structured text are excluded.
Ladder (Inline structured text)	Devices/labels in the selected line within the selected inline structured box	It follows the option setting for the ladder editor.
ST	Devices/labels in the selected line	—

Operating procedure

1. Set the following option.

[Tool] ⇨ [Options] ⇨ "Monitor" ⇨ "Ladder Editor"/"ST Editor" ⇨ "Setting for Automatic Registration to Watch Window"

2. During monitoring, select the range included in devices/labels to be registered to the Watch window.

Changing current values

The value can be entered directly in the "Current Value" column during monitoring.

For a bit device, select a row and double-click with pressing the Shift key or press the Shift+Enter keys to change the current value.

For S□, BL□\S□, BL□, steps/blocks are activated/deactivated by changing the current values. (☞ Page 273 Changing active status)

Precautions

When "Use File Register of Each Program" is selected in "File Setting" of the CPU parameter, the file register file that is available at END is changed. Therefore, changing the current values while specifying file registers for each program cannot be performed.

Importing to/exporting from file

Export/import the information displayed on the Watch window to/from a file.

Operating procedure

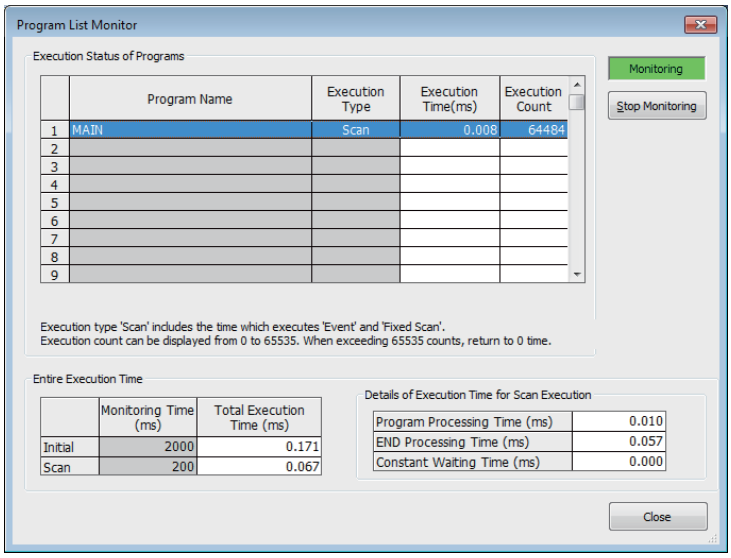
Select [Edit] ⇨ [Import File]/[Export to File].

14.5 Checking Processing Time of Program

Check the processing time of the program being executed using the Program List Monitor function.
FX5CPU does not support this function.

Window

[Online] ⇨ [Monitor] ⇨ [Program List Monitor]


Program List Monitor
Execution Status of Programs
Program Name
Execution Type
Execution Time(ms)
Execution Count
1 MAIN Scan 0.008 64484
Monitoring
Stop Monitoring
Execution type 'Scan' includes the time which executes 'Event' and 'Fixed Scan'.
Execution count can be displayed from 0 to 65535. When exceeding 65535 counts, return to 0 time.
Entire Execution Time
Monitoring Time (ms)
Total Execution Time (ms)
Initial 2000 0.171
Scan 200 0.067
Details of Execution Time for Scan Execution
Program Processing Time (ms) 0.010
END Processing Time (ms) 0.057
Constant Waiting Time (ms) 0.000
Close

Displayed items

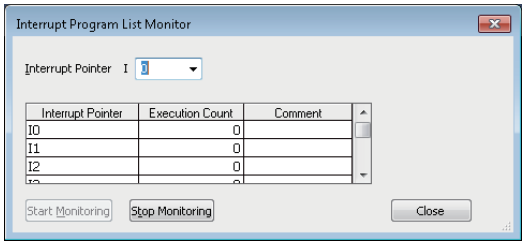
Item		Description
Execution Status of Programs	Execution Time (ms)	Displays the actual scan time (current value). The display contents are differ depending on the program execution type. At the program stop (standby) state: 0.000 ms
	Execution Count	Displays the number of executions, counted from '0' at the point of turning the CPU module to the RUN state. It is stored even after the program stops.
Entire Execution Time	Monitoring Time (ms)	Displays the program execution monitoring time set to "PLC RAS" of the CPU parameter.
	Total Execution Time (ms)	Displays the total scan time of the programs executed in the CPU module.
Details of Execution Time for Scan Execution	Program Processing Time (ms)	Displays the total execution time of scan execution type program.
	END Processing Time (ms)	Displays the END processing time.
	Constant Waiting Time (ms)	Displays the waiting time of the constant scan when the constant scan is set.

14.6 Checking Execution Counts of Interrupt Programs

Check the execution count of interrupt programs used in a program using the Interrupt Program List Monitor function.
FX5CPU does not support this function.

Window

[Online] ⇨ [Monitor] ⇨ [Interrupt Program List Monitor]



Operating procedure

Input the interrupt pointer number of which execution counts are to be displayed in "Interrupt Pointer".

Displayed items

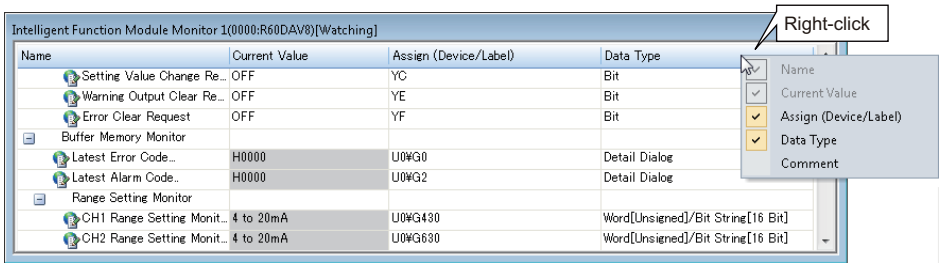
Item	Description
Execution Count	Starts counting when the CPU module turns to the RUN state and displays the execution counts. When reaching 65536, it returns to 0.

14.7 Checking Current Values in Intelligent Function Module

Check the I/O signals and the current values of buffer memory using the Intelligent Function Module Monitor function.
FX5CPU does not support this function.

Window

[View] ⇨ [Docking Window] ⇨ [Intelligent Function Module Monitor] ⇨ [Intelligent Function Module Monitor 1] to [Intelligent Function Module Monitor 10]



Operating procedure

1. Register an Intelligent function module to be monitored. (☞ Registering Intelligent function modules)
2. Select [Online] ⇨ [Watch] ⇨ [Start Watching].

"[Watching]" is added on the Watch window title during monitoring.

Point

Select and right -click the module information, select [Copy] from the shortcut menu to paste it to a text file.

Customizing monitoring items

Select and right-click the items on the Intelligent Function Module Monitor window, then select [Cut]/[Copy]/[Paste]/[Delete] from the shortcut menu to customize the items.
If the customized project is saved, the customized items will not be retained when the project is opened next time.
The customized items cannot be restored again.

Precautions

The cut/copied items can only be pasted to the same project in the same window.

■Detailed display of history information

Display the details of history information during monitoring.
Double-click a "Data Type" row in which "Detailed Dialog" is displayed, or right-click it and select [Detailed Dialog] from the shortcut menu.
The unsupported menus for the module cannot be used.

■Detailed display of error codes/alarm codes

Display the details of an error code/alarm code on the Intelligent Function Module Monitor windows or Detailed Dialog.
Double-click a row in which an error code/alarm code is displayed, or right-click it and select [Detail Display] from the shortcut menu.

■Customizing multiple Intelligent Function Module Monitor windows

Each axis of the positioning module can be monitored by activating multiple Intelligent Function Module Monitor windows and customizing each window.

Precautions

The Intelligent Function Module Monitor updates current values only within the range displayed on the window for monitoring acceleration.

When performing operation such as copy and paste on Excel®, "--" or the previously monitored current value is displayed for the data outside the range displayed on the window.

Registering Intelligent function modules

Register an Intelligent function module to be monitored in the Intelligent Function Module Monitor window.

Operating procedure

■Register Intelligent function modules using shortcut menu from Project view

1. Select an Intelligent function module to be registered in the Intelligent Function Module Monitor window from the Project view.
2. Right-click ⇨ select [Register to Intelligent Function Module Monitor] from the shortcut menu.

■Register Intelligent function modules by dragging and dropping from Project view

1. Select a module to be registered from the Project view.
2. Drag and drop it onto the Intelligent Function Module Monitor window.

■Register Intelligent function modules using shortcut menu from Intelligent Function Module Monitor window

1. On the Intelligent Function Module Monitor window, right-click and select [Register Module Information] from the shortcut menu.
2. Select the module to be registered in "Module List", and click the [OK] button.

When a positioning module is selected in "Module List", select the item displayed in "Monitor Item Category List".

Point

The module information can be copied and pasted to a file such as a text file by right-clicking the arbitrary module information and selecting [Copy] from the shortcut menu on the Intelligent Function Module Monitor window.

14.8 Saving and Checking Device Value

Check the device values of a CPU module at any timing using the Memory Dump function of the CPU module.
FX5CPU does not support this function.
By setting the trigger conditions, the data at a time when the conditions have been satisfied are collected and saved to the SD memory card. The saved device values can be checked with GX Works3.
For more details on the Memory Dump function of CPU module, refer to the following manual.
📖 MELSEC iQ-R CPU Module User's Manual (Application)

How to use the Memory Dump function

1. Set the internal buffer capacity of the CPU module to be used for the Memory Dump function. (☞ Page 284 Setting internal buffer capacity for the Memory Dump function)
2. Write the memory dump setting file. (☞ Page 285 Writing memory dump settings)
   Once it is written, the memory dump result file is saved to SD memory card when the set trigger conditions are satisfied.
3. Read the memory dump result file to the personal computer. (☞ Page 286 Reading memory dump results)
4. Check the read memory dump result file. (☞ Page 287 Displaying memory dump results)

Setting internal buffer capacity for the Memory Dump function

Set the internal buffer memory capacity for the Memory Dump function using "Internal Buffer Capacity Setting" in "Memory/Device Setting" of the CPU parameter. Write the parameter to the CPU module after the setting is configured.
However, set the capacity within the range that meets the following conditions when the Memory Dump function is used in conjunction with the data logging function and the realtime monitor function of CPU module.
The total of data logging buffer capacities + Memory dump buffer capacity + Realtime monitor buffer capacity ≤ 3072KB
Each buffer capacity can be checked with the following:

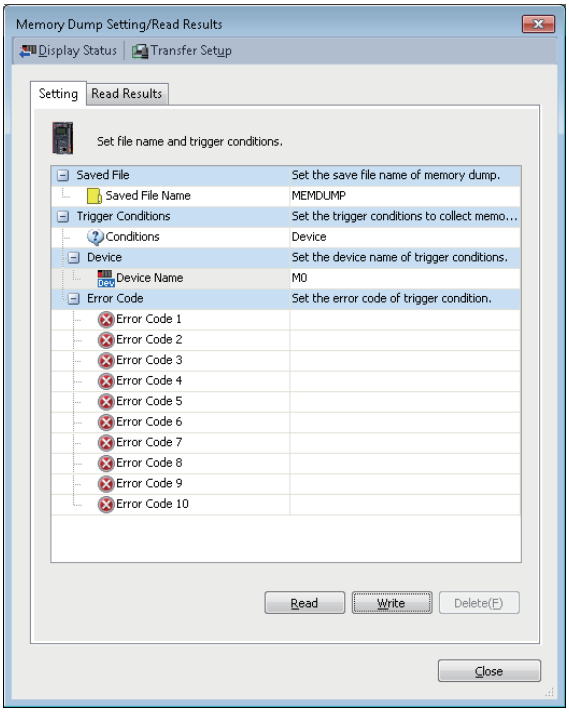
- Data logging buffer capacity, Memory dump buffer capacity: CPU parameter
- Realtime monitor buffer capacity: The realtime monitor setting of GX LogViewer

Writing memory dump settings

Once the memory dump setting file is written, the Memory Dump function is started and waits for a trigger.
If the memory dump setting file has already been stored, the Memory Dump function is started by powering ON the CPU module.

Window

[Debug] ⇨ [Memory Dump] ⇨ [Setting]



Operating procedure

Set the items and click the [Write] button.

Item	Description
Saved File	Set a name of a file to be output when the Memory Dump function is performed. A number, '00' to '99' is added to the end of the set name for the file name to be output actually.
Trigger Conditions	Specify the trigger conditions to collect memory dump. • Device: Specify this to set the timing when the bit data of a specified device turns OFF → ON as the trigger condition. • Error Code: Specify this to set an error code of CPU module as the trigger condition. • Device/error Code: Specify this to set a device/error code as the trigger condition. The Memory Dump function is performed when any of multiple conditions is satisfied.
Device	Specify a device to be set as the trigger condition. Only global devices can be specified. ■Device representation that can be specified • Bit device: X, Y, M, L, F, SM, V, B, SB, T (enter 'TS'), ST (enter 'SS/STS'), C (enter 'CS'), LT (enter 'LTS'), LST (enter 'LSS/LSTS'), LC (enter 'LCS'), FX, and FY • Word device (bit specification): D, SD, W, SW, R, ZR, FD, RD
Error Code	Specify a 4-digit error code to be set as the trigger condition in hexadecimal. The wild card, "*" can be specified for the 1st digit of an error code only.

Point

The number for annunciator cannot be specified as the error code to be set as the trigger condition. Use the device name in order to specify the number of annunciator.

Considerations for memory dump setting

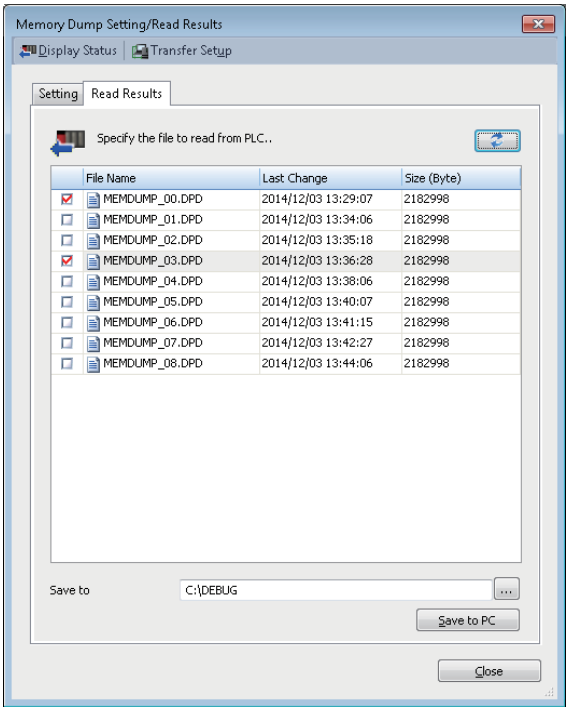
- Before writing the memory dump setting, check that the trigger condition has not already satisfied.
- When a file register is specified as the trigger condition, do not change the file register file name and the file register block No. once the memory dump setting has been written.

Reading memory dump results

To check the memory dump result file with GX Works3, read the memory dump file from SD memory card inserted in the CPU module and save it to the personal computer in advance.

Window

[Debug] ⇨ [Memory Dump] ⇨ [Read Results]



The current execution status of the Memory Dump function can be checked by selecting [Display Status].

Operating procedure

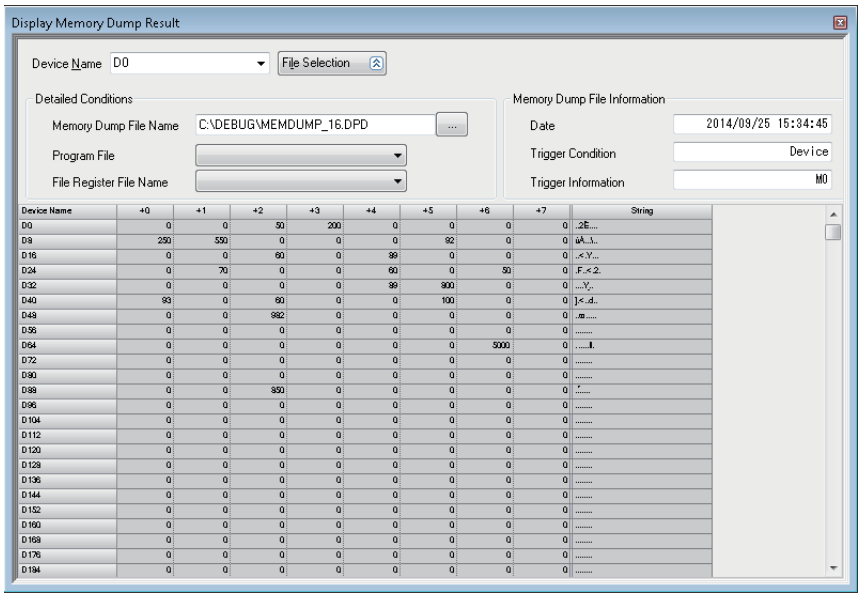
1. After clicking , specify the memory dump result file to be read.
2. Specify "Save Destination", and click the [Save to PC] button.

Displaying memory dump results

Display a memory dump result file save in the personal computer.

Window

[Debug] ⇨ [Memory Dump] ⇨ [Display Result]



Operating procedure

Enter the path for the memory dump result file to be displayed in "Memory Dump File Name".

- To specify a local device for "Device Name", select the program file to be referred in "Program File".
- To specify 'R' or 'ZR' for "Device Name", select the file register to be referred in "File Register File Name". When 'R' is specified, the device for the first block is displayed.
- To specify 'Z' or 'LZ' for "Device Name", select the program file to be referred in "Program File".

14.9 Checking Sampled Data on Program Editor

Use the offline monitor function to check the sampled data (memory dump result file) from a CPU module on the program editor.

By using this function, it is possible to reproduce the sampled data on the program editor from a remote location and investigate the causes when error occurs.

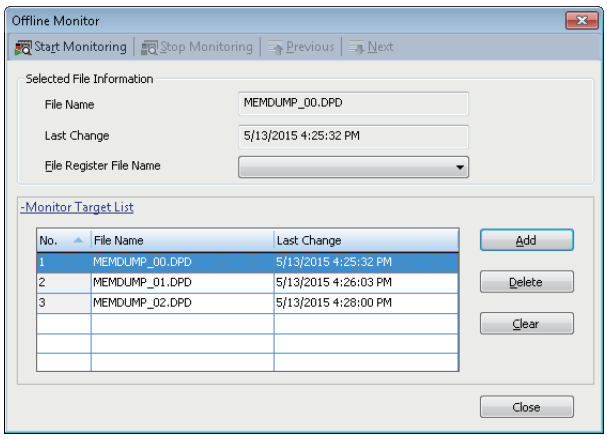
Open the program editor to be monitored in advance.

For the methods for sampling data, refer to the following section.

☞ Page 284 Saving and Checking Device Value

Window

[Debug] ⇨ [Offline Monitor]



Precautions

If the project that outputs the memory dump result file differs from the one for executing the offline monitor function, the device value may not be obtained.

- Same project: Since the number of device points are same, the device to be monitored exists in the memory dump result file.
- Different project: When the device to be monitored does not exist in the memory dump result file, the value is displayed as an indefinite value.

■Online operation while the offline monitor function has been executed

While the offline monitor function has been executed, the following online functions are available.

- Monitor on a program editor
- Device/Buffer Memory Batch Monitor
- Watch

Executing the offline monitor function

Operating procedure

1. Click the [Add] button on the "Offline Monitor" screen.
2. Select the memory dump result file on the "Open File" screen, then click the [Open] button. (Multiple selections allowed.)
3. Select the file to be monitored from the list on the "Offline Monitor" screen, then select [Start Monitoring].

Monitoring the files is executed on the program editor. It is also executed on the Watch window and the "Device/Buffer Memory Batch Monitor" screen.

■Switching the target file

If the target file is switched by selecting [Previous]/[Next] while the offline monitor function has been performed, the device values are updated to the one of the selected file.

Stopping the offline monitor function

Operating procedure

Select [Stop Monitoring] on the "Offline Monitor" screen.

Monitoring is stopped on the program editor. It is also stopped on the Watch window and the "Device/Buffer Memory Batch Monitor" screen.

Considerations for monitoring files

■When the device value of monitor target cannot be obtained

If the device value to be monitored does not exist in a selected file, the value will be displayed corresponding to the device type as follow.

Since the indirectly specified devices are not supported, they are also displayed such as the following.

- Bit device, bit-specified word device: FALSE (0)
- Word device, double-word device, FLOAT [Double Precision]: -1

■Active status of blocks/steps in an SFC program

Active status is not displayed on the SFC diagram editor.

All blocks are displayed as "Uncreated Block" on all SFC blocks batch monitor.

MEMO

PART 5 MAINTENANCE AND INSPECTION

This part explains the maintenance and inspection methods such as a function to protect data (prevent from manipulation and data leakage) and display the system status of the CPU module.

15 PROTECTING DATA

16 DIAGNOSING MODULE

17 SAMPLING DEVICE DATA

18 CHECKING/CHANGING CPU MODULE OPERATION

15 PROTECTING DATA

This chapter explains how to protect the data in a project.

The functions to save the data are as follows:

Purpose	Target	Function name	Reference
Prevents illegal access to programs (in POU unit). (Using password)	Project	Block password function	Page 293 Preventing Illegal Access to Program (Protected by Password)
Prevents illegal access to programs (in program file unit). (Using security key)	Project	Security key authentication function	Page 295 Preventing Illegal Access to Program (Protected by Key)
Prevents illegal execution of programs. (Using security key)	CPU module		Page 300 Preventing Illegal Program Execution
Prevents illegal access to projects. (Using password)	Safety project	User Authentication	Page 303 Preventing Illegal Access to Project
Prevents illegal access to CPU module. (Using password)	CPU module (RnSFCPU only)		Page 306 Preventing Illegal Access to CPU Module
Prevents illegal read/write operation for files. (Using password)	CPU module	File password function	Page 307 Preventing Illegal Data Reading/Writing
Restricts access from other than specific communication route. (Using password)	CPU module	Remote password function	Page 310 Restricting Access from Other Than Specific Communication Route
Identifies the IP address of external devices via Ethernet to block access from an illegal IP address. (Setting with module parameters)	CPU module	IP filter function	MELSEC iQ-R Ethernet User's Manual (Application)

15.1 Preventing Illegal Access to Program (Protected by Password)

To prevent an illegal access to programs (in POU unit) by setting a password, use the Block password function.

Block password function

Restrict the operations by registering a password to POUs.

- The operation that can be restricted: Accessing to POU (The operations will be allowed while the program is authenticated even if a password has been registered.)
- Target data: Program blocks, function blocks, and functions (☞ Page 58 Data configuration)

How to use block password

Procedure to enable security

1. Set a block password to a POU. (☞ Page 293 Setting block password)
2. Save the project. (☞ Page 69 Saving project)

After closing the project in which the block password is set, the POU cannot be accessed when the file is opened next time.

Procedure to make locked POUs accessible

- Authenticate the block password set to the POU. (☞ Page 294 Authenticating block password)

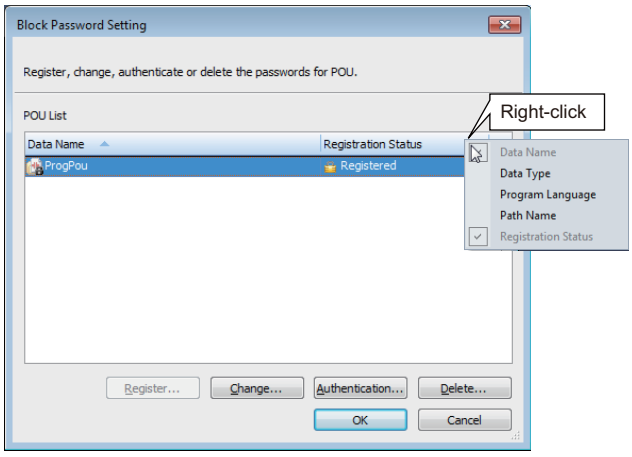
The POU can be accessed while the project is open.

Setting block password

Register a block password to POUs.

Window

- [Project] ⇨ [Security] ⇨ [Block Password Setting]
- Select a POU on the Project view, then right-click and select [Block Password Setting] from the shortcut menu.



Registering/changing block password

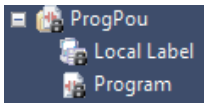
Register/change a block password to POUs.

Operating procedure

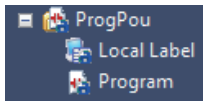
1. Select a POU to set a block password and click the [Register]/[Change] button.
2. Enter the password on the "Register Password" screen/"Change Password" screen, and click the [OK] button.

When a block password is registered, the icons are displayed on the Project view as shown below.

Registered



Authenticated



Point

By selecting multiple POUs in "Data Name", a block password can be registered to the selected POUs in batch.

Authenticating block password

Unlock the POU temporarily by authenticating the password which has been set to it.
Once the password is authenticated, the data can be accessed until the project is closed.

Operating procedure

1. Select the POU to authenticate the block password, and click the [Authentication] button.
2. Enter the password on the "Password Authentication" screen, and click the [OK] button.

Deleting block password

Delete the block password registered to a POU.

Operating procedure

1. Select the POU to delete the block password, and click the [Delete] button.
2. Enter the password on the "Delete Password" screen, and click the [OK] button.

15.2 Preventing Illegal Access to Program (Protected by Key)

To prevent an illegal access to programs (in program file unit) with a key, use the Security key authentication function.

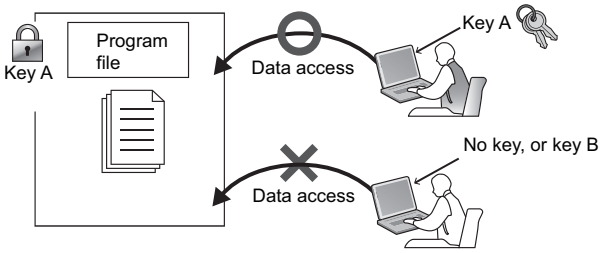
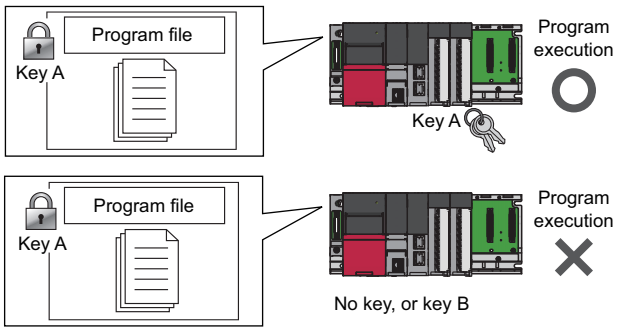
Security key authentication function

The operations can be restricted by using a security key.

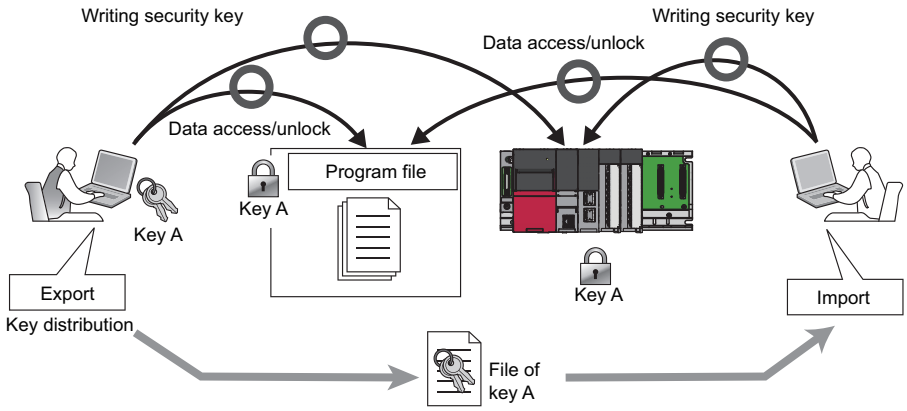
Security key setting target:

Target	Purpose
■Program files (Program file, FB file, FUN file) (☞ Page 58 Data configuration) A security key can be registered in program file unit, however, all the program files in the project will be the same security key.	• Restrict the access for program files in the project. • Restrict the execution of programs by CPU module.
■Personal computer Multiple security keys can be registered in one personal computer.	Enable the access for the program files protected by a security key.
■CPU module Only one security key can be written to one CPU module.	Enable the execution of the program restricted by a security key.

The following operations can be restricted:

Access to programs (Program file, FB file, FUN file)	Execution of programs in CPU module (STOP/PAUSE → RUN)
Restrict the operations by detecting mismatches of the security key between the program file and personal computer. 	Restrict the operations by detecting mismatches of the security key between the program file written in a CPU module and CPU module.  The programs are not executed if any of the programs of which security key is unmatched in the CPU module exist. The security key written in the CPU module is retained after powering OFF.
☞ Page 296 How to use the Security key authentication function (access prevention)	☞ Page 300 How to use the Security key authentication function (execution prevention)

Copy of security key: The security key registered to a personal computer can be copied by exporting/importing it from/to other personal computer. In addition, the expiration date of the security key can be set.



Setting method: ☞ Page 298 Copying security key

Precautions

The security key registered to a personal computer is not deleted even if GX Works3 is uninstalled. Delete the security key on the "Security Key Management" screen.

How to use the Security key authentication function (access prevention)

Procedure to enable security

1. Create a security key. (☞ Page 297 Creating/deleting security key)
2. Register the security key in a program file. (☞ Page 299 Registering security key in program file)
3. Save the project. (☞ Page 69 Saving project)

A personal computer which does not have the same security key as the one registered in the project cannot be accessed.

Procedure to make locked program files accessible

1. Copy the same security key as the saved program file. (☞ Page 298 Exporting security key)
2. Register the security key in a personal computer from which a program file is to be accessed. (☞ Page 298 Import)
3. Open the program file. (☞ Page 63 Opening project)

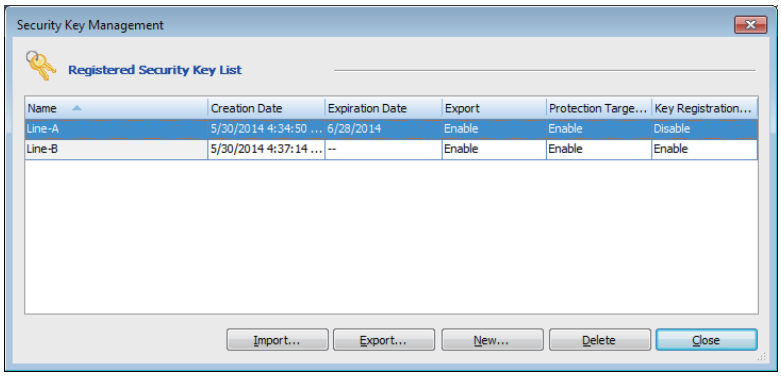
A program file can be accessed when the security key between the program file and a personal computer is consistent.
The operations shown above are not required for the personal computer with which a security key is registered in a program file. However, the above operations are required when the security key is deleted.

Creating/deleting security key

Create/delete security keys on the "Security Key Management" screen.

Window

[Project] ⇨ [Security] ⇨ [Security Key Management]



Displayed items

Item	Description
Creation Date	Displays the registered date of the security key.
Expiration Date	Displays the expiration date set when the security key was exported.
Export	Displays "Enable/Disable" that indicates whether re-exporting the security key is enabled/disabled set when the security key was exported.
Protection Target Selection	Displays "Enable/Disable" that indicates whether selecting data to be protected is enabled/disabled set when the security key was exported.
Key Registration to CPU	Displays "Enable/Disable" that indicates whether writing data to CPU module is enabled/disabled set when the security key was exported.

Creating security key

Create a security key and register it to the personal computer.

Number of security key registration: The security keys can be created for each logon user. Up to 128 security keys combined with the created security keys and imported ones can be registered for each logon user.

Operating procedure

1. Click the [New] button.
2. Enter the security key name on the "New Security Key" screen and click the [OK] button.

Deleting security key

Delete the security key registered to the personal computer.

If the security key used to lock the project is deleted, the locked data in the project cannot be accessed/edited.

After deleting the security key, even when the security key with the same name as the deleted one is recreated, it will not be the same security key as the deleted one. Take extra caution when deleting a security key.

Operating procedure

1. Select the security key to be deleted on the "Security Key Management" screen.
2. Click the [Delete] button.

Copying security key

Copy (import/export) security keys on the "Security Key Management" screen.

Exporting security key

Export the security key registered to the personal computer to the file format (*.ity) which can be imported.
Expiration date and restrictions for operations can be added to the security key to be exported.

Operating procedure

1. Select the security key to be exported on the "Security Key Management" screen.
2. Click the [Export] button.
3. Set the following items on the "Export" screen.

Item	Description
Restriction	Set whether to add restrictions to the security file to be exported.
Expiration Date	Set the expiration date for the security key to be exported.
Export	Set whether to allow exporting the security key after importing the exported security key.
Protection Target Selection	Set whether to allow registering the security key for the program files using the exported security key.
Key Registration to CPU	Select whether to allow writing the security key to the CPU module using the exported security key.

4. Set the password to be required when importing the security key, and click the [Export] button.

Precautions

- The exported security key files should be tightly controlled.
- The security key files created with GX Works3 are not compatible with that of GX Works2.

Import

Import the exported security key file in a personal computer, and register the security key.

Operating procedure

1. Click the [Import] button on the "Security Key Management" screen.
2. Select a file (*.ity) on the "Import Security Key" screen, and click the [Open] button.
3. Enter the password set when the security key was exported on the "Password Authentication" screen, and click the [OK] button.

■Expiration date

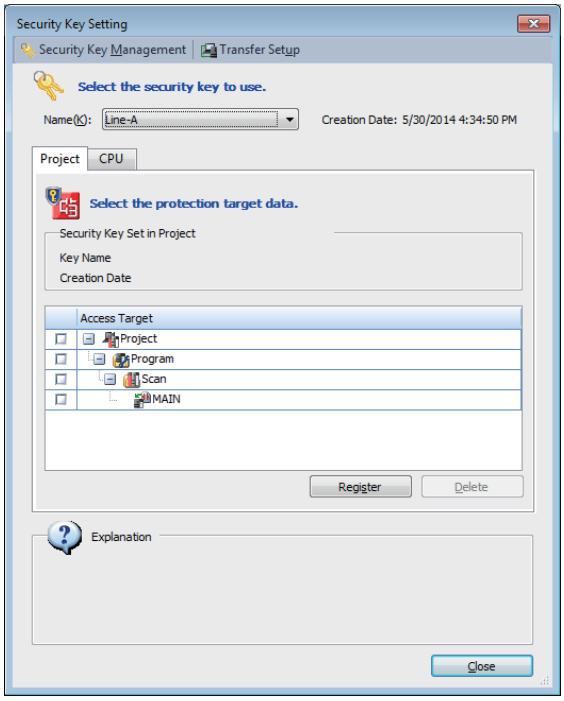
- If the expiration date of the security key registered in a personal computer is expired, accessing programs, re-exporting security keys, registering security keys in program files, and writing security keys to CPU module cannot be performed.
- To use the same security key after the expiration date, export the security key from the exported personal computer again, and import it to the personal computer.

Registering security key in program file

Create a security key on the "Security Key Management" screen in advance.
A security key can be registered in program file unit, however, the same security key will be set to the program files in the project.

Window

[Project] ⇨ [Security] ⇨ [Security Key Setting]



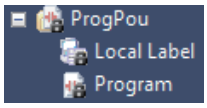
Registering security key

Operating procedure

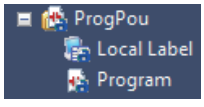
1. Select a security key to be registered to a program file from "Name".
2. Select the program file to lock from the [Project] tab, and click the [Register] button.

The security key registered in program files will be validate after saving the project.
The program file of which security key is registered is displayed in the Project view as follows:

When the security keys between in the personal computer and the project does not match



When the security keys between in the personal computer and the project matches



■Security key lock for FB files

If the program, in which the function block of FB file locked with a security key is used, is opened on the personal computer with no corresponding security key exists, the FB program cannot be opened.
However, the instances of the locked FB can be created in the program on the personal computer in which the security key is not registered.

Deleting security key

Operating procedure

Click the [Delete] button in the [Project] tab.

15.3 Preventing Illegal Program Execution

To prevent an illegal execution for programs (STOP/PAUSE → RUN) written in a CPU module, use the Security authentication function.

For details on the Security key authentication function, refer to the following section.

☞ Page 295 Security key authentication function

How to use the Security key authentication function (execution prevention)

Procedure to enable security

1. Create a security key. (☞ Page 297 Creating/deleting security key)
2. Register the security key in the program file to be written to the CPU module. (☞ Page 299 Registering security key in program file)
3. Write the program files to the CPU module. (☞ Page 250 Writing data to programmable controller)

A CPU module which does not have the same security key as the one registered in the program file cannot execute the programs.

Procedure to make locked programs executable

1. Set the Specify Connection Destination to access to the CPU module. (☞ Page 235 Specification of Connection Destination)
2. Write the same security key as the protected program file to the CPU module. (☞ Page 301 Writing/deleting security key to/from CPU module)

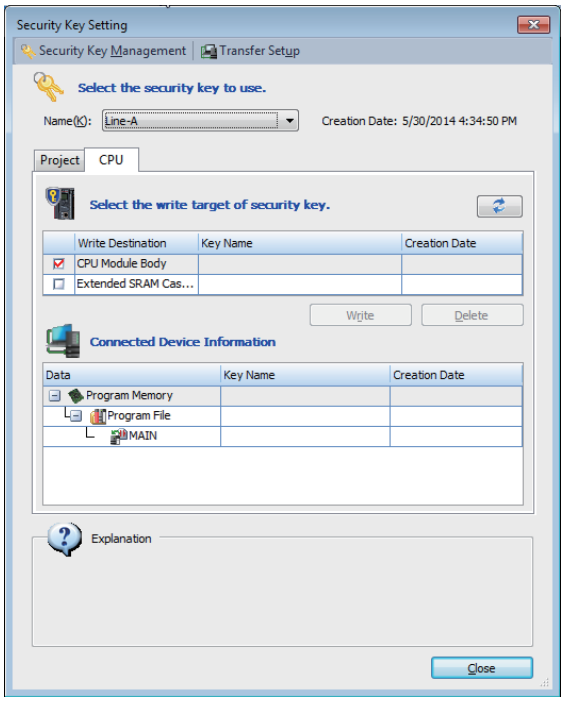
The programs can be executed unless otherwise the security key set to the program file or CPU module is changed.

Writing/deleting security key to/from CPU module

By writing the same security key to the CPU module, execution of the programs can be permitted.
Only one security key can be written to either a CPU module or an extended SRAM cassette.
FX5CPU does not support an extended SRAM cassette.
Additionally, one security key can be written to multiple CPU modules.
When writing a security key, create a key on the "Security Key Management" screen and connect a personal computer and CPU module in advance. (Writing/deleting can be performed when a CPU module is n 'STOP' state only.)
For a safety project, the security key can be written/deleted when the safety operation mode of RnSFCPU is in the test mode.

Window

[Project] ⇨ [Security] ⇨ [Security Key Setting]



Writing security key

Operating procedure

1. Select the security key to be written from "Name".
2. Select the target to be written in the [CPU] tab, and click the [Write] button.

■When a security key is written to an extended SRAM cassette

When replacing the CPU module, the security key can be succeeded to the replaced CPU module only by replacing the extension SRAM cassette. Therefore, writing the security key again from the personal computer in which the security key is registered is unnecessary.

■For multiple CPU system configuration

When setting a security key for multiple CPU system configuration, write the security key to each CPU module. In addition, when using the same security key as CPU No.1 in CPU No.2 to CPU No.4 , write the same security key as CPU No.1 to each CPU module.

Deleting security keys

The security key written in the CPU module can be deleted on the personal computer in which the security key is not registered.
However, the program of which security key is registered in the CPU module exists, the security key cannot be deleted.

Operating procedure

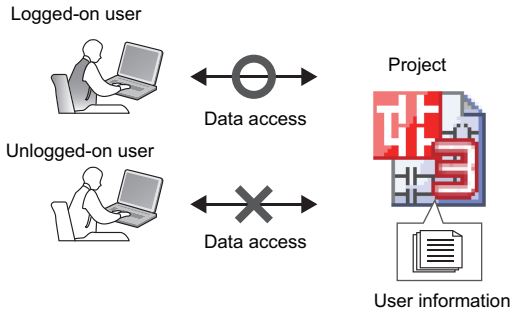
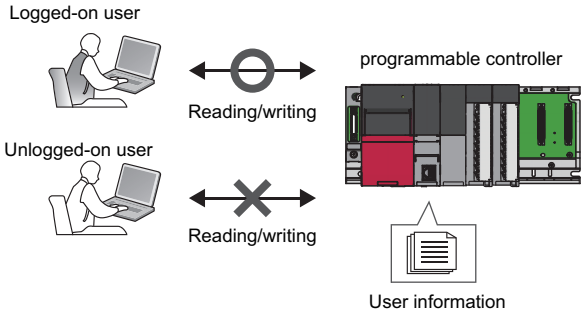
Select the security key to be deleted from the [CPU] tab, then click the [Delete] button.

15.4 Preventing Illegal Access to Project

To prevent an illegal access to a safety project from the users with no authority, use the User Authentication function.
Only RnSFCPU supports the User Authentication function.

User Authentication function

The operations can be restricted by registering a user/password in a project and a CPU module.
The following operations can be restricted:

Access to project	Access to CPU module
To access to a project, logging on with the registered user information (user name/password) is required.	To access to a CPU module, logging on with the registered user information (user name/password) is required.
	
☞ Page 304 How to use the User Authentication function (access prevention)	☞ Page 306 How to use the User Authentication function (access prevention)

The user information registered in a CPU module needs to match with the user information of a project.

Access level

An access level is an operation authority given to a user who logs on to a project/CPU module.

Access level		Operation authority
Higher	Administrators	■Administrator level All operations, such as user management of projects and CPU modules, are performable.
↕	Developers	■Developer level Operations except for the user management and the security setting are performable.
Lower	Users	■Operator level Referring to a project and monitoring of CPU modules are performable.

■Availability of project function (Operations required the user authentication)

The following table shows the functions restricted in projects depending on the access level.
○: Available, ×: Not available

Function	Access level		
	Administrators	Developers	Users
Save	○	○	×
User management	○	×	×
Delete user data of project	○	×	×
Project revision	○	○	×
Change module type	○	×	×
Online program change	○	○	×

■Availability of CPU module function (Operations required the user authentication)

The following table shows the functions restricted in CPU module depending on the access level.

○: Available, ×: Not available

Function		Access level		
		Administrators	Developers	Users
File operation	Write to PLC	○*1	○*1	×
	Read from PLC	○*1	○*1	○*1
	Verify with PLC	○*1	○*1	○*1
	Delete data in a programmable controller	○*1	○*1	×
CPU memory operation	Initialization/Clear Value	○	○	×
Change program	Online Program Change	○	○	×
	File batch online change	○	○	×
Security	Write/delete a security	○	×	×
	Register/change/delete file password	○	×	×
	Authenticate file password	○	○	○
	Add/delete/change new user	○	○	×
	Read/write user data	○	×	×
Diagnostics	All clear event history	○	○	×
Safety function	Switch Safety Operation Mode	○	○	×

*1 User authentication is not required to write/read a device memory (except for the file register).

How to use the User Authentication function (access prevention)

1. Create a safety project, and add a user of Administrators to the project. (☞ Page 61 Creating project)
2. Add the users who are allowed to operate the project. (☞ Page 305 User management)
3. Save the project. (☞ Page 69 Saving project)

Once the above procedure has done, logging on is necessary to access to the projects.

Logging on to project

For the project that the user information is registered, logging on is required to perform the operations such as opening a project.

Enter the user name and the password, that are registered in the project, to log on.

Changing password of logon user

Change the password of the user who is logging on to a project.

Operating procedure

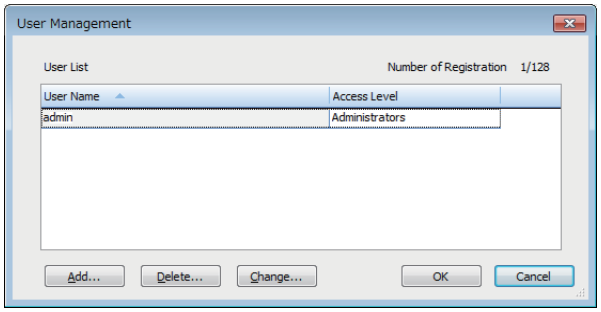
Select [Project] ⇨ [Security] ⇨ [Change User Password].

User management

Manage the user registration status of a project to which the security is set.
A User of the Administrators can change all user's passwords.

Window

[Project] ⇨ [Security] ⇨ [User Management]



Adding/changing users

Add/change a user information (user name/password) to/from a project.

Operating procedure

1. Click the [Add]/[Change] button.
2. Set the items on the "Add New User" screen/"Change User" screen, and click the [OK] button.

Deleting users

Delete the user information registered in a project.

Operating procedure

Select the user name to be deleted, and click the [Delete] button.

15.5 Preventing Illegal Access to CPU Module

To prevent an illegal access to the project written to a CPU module, use the User Authentication function.
Only RnSFCPU supports the function.
For details on the User Authentication function, refer to the following section.
☞ Page 303 User Authentication function

How to use the User Authentication function (access prevention)

1. Register the user information in the project which is to be written to a CPU module. (☞ Page 305 User management)
2. Write the user information to a CPU module. (☞ Page 306 Writing/reading user information to/from CPU module)
3. Write programmable controller data to a CPU module. (☞ Page 250 Writing data to programmable controller)

Once this procedure has done, logging on is required to access to the CPU module in advance.

Writing/reading user information to/from CPU module

Write the user information registered in a project to a CPU module.
Read the user information registered in a CPU module, and overwrite the existing project.

Operating procedure

Select [Online] ⇨ [User Authentication] ⇨ [Write User Data to PLC]/[Read User Data from PLC].

Logging on to CPU module

Logging on is required to access to the CPU module in which the user information is registered.
Enter the user name and the password, that are registered in the CPU module, to log on.
After logging on, the user will be logged off automatically if the CPU module has no access for five minutes (operations required the user authentication) or was written the user information.

Operating procedure

1. Select [Online] ⇨ [User Authentication] ⇨ [Log on to PLC]
2. Set the items on the "User Authentication" screen, and click the [OK] button.

■When a user name and a password used in the User Authentication function are lost

Initialize the user information, the programmable controller data, and the security keys in the CPU module by performing the following operation.

- Select [Online] ⇨ [User Authentication] ⇨ [Initialize all PLC Data]

Rewrite the user information, the programmable controller data, and security keys (optional) to the CPU module after initialization.
If the power of the CPU is turned OFF → ON or the CPU module was reset, the allocation for the device/label memory is back to the initial status. (Each area has the default capacity.)

Changing password

Change the password for the CPU module to the same password set to a project.

Operating procedure

Select [Online] ⇨ [User Authentication] ⇨ [Change the Password of PLC]

15.6 Preventing Illegal Data Reading/Writing

To prevent data from being illegally read from/written to CPU module, use the File password function.

File password function

By registering a password to the files in the CPU module, the password authentication is required when reading/writing files.
For FX5CPU, selecting "Permanent PLC Lock" will enables to keep the operation to read/write files prohibited and to prevent it from being canceled. Setting a password is not required when "Permanent PLC Lock" is selected.
There are two kinds of protections in password and "Permanent PLC Lock"; "Read Protection" and "Write Protection". Both of them can be set to one file.

■ Operations that require password authentication by setting a file password

○: Required, —: Not required

Online operation	Applicability	Password to be set
Read to a programmable controller	○	Read-protected password
Write to a programmable controller	○	Write-protected password
Online program change (Online Program Change, File batch online change)	○	Write-protected password
Verification with programmable controller	○	Read-protected password
File password setting (register/change/authentication/deletion)	○	Read-protected password, write-protected password
Delete data in a programmable controller	○	Write-protected password
CPU memory operation (initialization)	—	—

■ Target file

○: Available, ×: Not available

File name	Availability
System parameter, CPU parameter, module parameter, module extended parameter, memory card parameter	○
Remote password	×
Global label setting	○
Global label assignment information	×
Global label initial value	×
Local label initial value	×
Program file	○
POU (FB file/FUN file)	○
File register	×
Device initial value	○
Common device comment	○
Each program device comment	○
Firmware update prohibited file	○

■When accessing from external devices using function other than GX Works3

A file password is validate when accessing a file using the FTP server function. For details, refer to the following manual.

📖 MELSEC iQ-R Ethernet User's Manual (Application)

A file password is validate when accessing a file using MC protocols. For details, refer to the following manual.

📖 MELSEC iQ-R Serial Communication Module User's Manual (Application)

Precautions

- Keep passwords in a secure place.
  If the registered password is lost, or "Permanent PLC Lock" set in FX5CPU is needed to be cancelled, initialize the CPU module using the CPU Memory Operation function (☞ Page 330 Checking Memory Usage) and write the project to the CPU module.
- To set "Permanent PLC Lock" to both of "Read Protection" and "Write Protection", configure the both settings at the time of the first registration. If only one of them was registered, the other cannot be registered later.

How to use file password

Procedure to enable security

1. Set the Specify Connection Destination to access to the CPU module. (☞ Page 235 Specification of Connection Destination)
2. Set a file password to the file in the CPU module. (☞ Page 309 Setting file password)

If the project in which the file password is set is closed, the file cannot be read and write when GX Works3 is started up next time.

Procedure to make locked files readable/writable

1. Set the Specify Connection Destination to access to the CPU module. (☞ Page 235 Specification of Connection Destination)
2. Authenticate the file password which has been set to the file in the CPU module. (☞ Page 309 Setting file password)

The file can be write and read while the project is open.

The files can also be read/written when entering a correct password in the "File Password Setting" screen displayed when accessing a file.

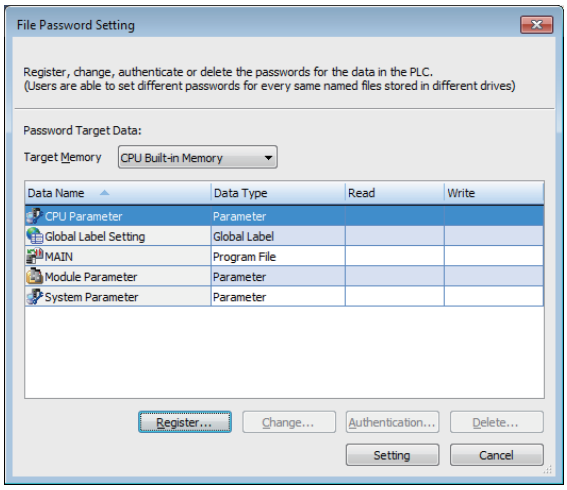
Setting file password

When registering a file password, connect a personal computer and CPU module in advance.
"Register", "Change", "Authentication", or "Delete" can be set on the setting screen of file password.
For a safety project, a file password can be set when the safety operation mode of RnSFCPU is in the test mode. Only the authentication can be performed in the safety mode.

Setting	Operation
Registration	Restrict online operations by setting password to data in the CPU module. A password set to data in the transfer source memory of boot operation remains set in the transfer destination memory.
Change	Change the password set to data in the CPU module.
Authentication	Unlock the data in the CPU module temporarily by authenticating the password. By authenticating a password, the corresponding data in the CPU module can be read/written until the project is closed.
Deletion	Delete a password set to data in CPU module.

Window

[Project] ⇨ [Security] ⇨ [File Password Setting]



Operating procedure

1. Select the target data and click the [Register]/[Change]/[Authentication]/[Delete] button.
2. Set the items on the displayed screen, and click the [Completed] button.
3. Click the [Setting] button on the "File Password Setting" screen.

15.7 Restricting Access from Other Than Specific Communication Route

To restrict the access from other than specific communication route, use the Remote password function.

Remote password function

By setting a password for the CPU module, accessing via the specified RJ71EN71, serial communication module, and built-in Ethernet CPU is limited.
For the modules that support remote password setting and details on the remote password setting, refer to the user's manual used.

How to use remote password

Procedure to enable security

1. Set the remote password. (☞ Page 311 Setting remote password)
2. Specify "Remote Password" and write the data to the CPU module. (☞ Page 250 Writing data to programmable controller)

The access to the CPU module will be restricted.

Procedure to access CPU module locked by the security function

Entering the password is required in order to access the CPU module. The access is permitted when the entered password is correct.

1. Access to the CPU module (writing or reading operation). (☞ Page 250 Writing data to programmable controller, Page 254 Reading data from programmable controller)
2. Unlock the remote password by following the message. (☞ Page 311 Setting remote password)

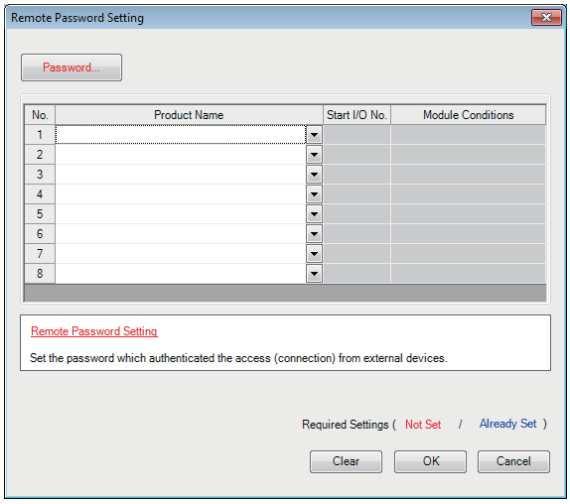
When the entered password is correct, the access to the CPU module will be permitted.

Setting remote password

The remote password can be set for up to 8 modules. (Up to 8 CPU modules for a multiple CPU configuration)

Window

Project view ⇨ "Parameter" ⇨ "Remote Password"



Operating procedure

1. Click the [Password] button.
2. Set the items on the "Register Password" screen, and click the [OK] button.

Item	Description
Product Name	Select the module of which remote connection is to be enabled. For CPU modules integrated network module (such as RnENCPU), select the followings to use the CPU part and the network part as a separate module. • CPU part: CPU Module (Built-in Ethernet Function) • Network part: CC-Link IE Built-in Ethernet I/F Module
Start I/O No.*1	Enter the start I/O number*1. (The specification using module labels is applicable.)
Module conditions	Set the remote password to enable/disable for the user connection number or system connection.

*1 For FX5CPU, "Intelligent Module No." is displayed.

16 DIAGNOSING MODULE

This function displays the state of modules, networks, and whole system. At an error occurrence, it displays an error content and corrective action.

The diagnostic functions of GX Works3 are as follows:

Function	Reference
System monitor*1	Page 313 Checking Module Status of System
Diagnosing CPU module	Page 314 Diagnosing CPU Module
Diagnosing network	Page 316 Ethernet diagnostics
	Page 317 CC-Link IE Controller Network diagnostics
	Page 319 CC-Link IE Field Network diagnostics
	Page 320 MELSECNET diagnostics
	Page 321 CC-Link diagnostics

*1 Not supported by FX5CPU.

16.1 Checking Module Status of System

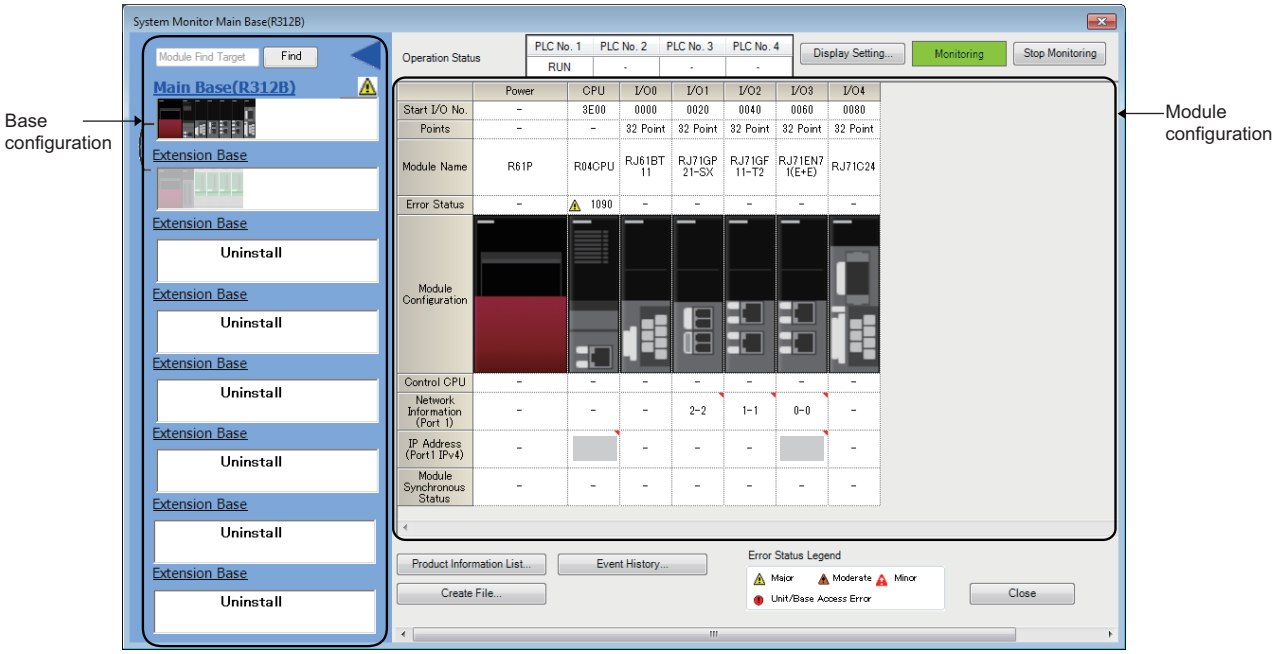
The System Monitor function displays the module configuration of the system in operation and detailed information of each module.

In addition, the error status can be checked and the module in which the error occurred can be diagnosed.

FX5CPU does not support this function.

Window

[Diagnostics] ⇨ [System Monitor]



- For Q series power supply, the model name is displayed as "Power" instead of their model name. Additionally, when using Q series module, the network information and IP address are not displayed.
- To display the network information and IP address of the port 2 of the modules who have two ports, click the [Display Setting] button.

■When '?' appears on an image of module

'?' appears on an image of a module when a connected module is in the following situation.

- A connected module is broken.
- Mounting status is different from the one set in I/O Assignment Setting of the System Parameter.
- A control CPU is not mounted.
- Its profile is not registered.

Precautions

The base unit configuration is not displayed in order of the connection if the number of extension base was set incorrectly.

Automatic diagnosis function

This function starts up the System Monitor function automatically when connecting CPU module and a personal computer by USB connection.

Set whether to start up automatically on the task tray.

Check of module firmware version and product information

The firmware version and product information can be checked by pressing the [Product Information List] button on the "Product Information List" screen.

16.2 Diagnosing CPU Module

This section explains how to display the status and error codes of modules.
FX5CPU only supports the following 'Diagnosing CPU modules'.

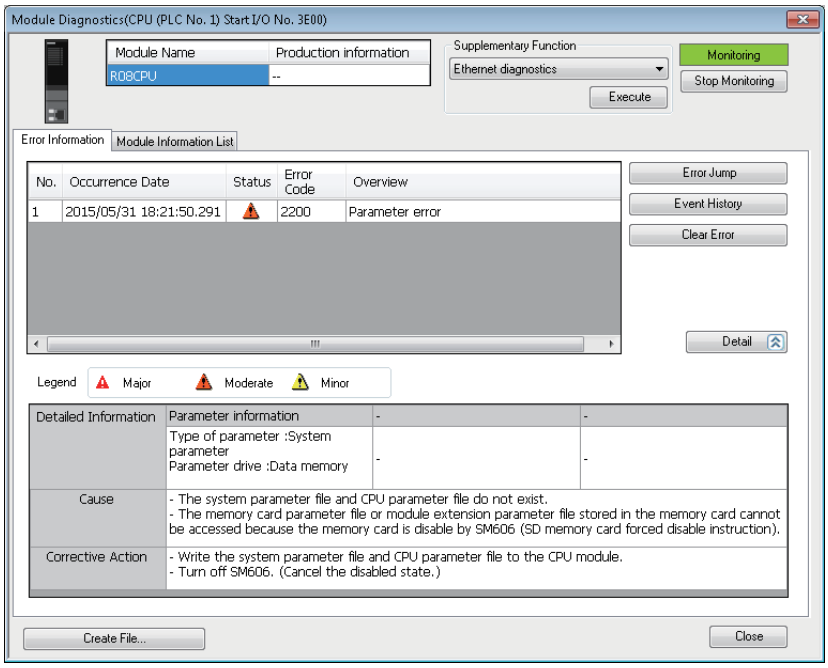
Window

■Diagnosing CPU modules

[Diagnostics] ⇨ [Module Diagnostics (CPU Diagnostics)]

■Diagnosing I/O modules and Intelligent function modules

Select the module to be diagnosed from the module configuration of the "System Monitor" screen and double-click one of the cells on the same column to display the "Module Diagnostics" screen.



For Q series module, '-' is displayed in Occurrence Date, Status, and Overview.

Precautions

Since the occurrence date depends on the time zone selected in "Clock Related Setting" for "Operation Related Setting" of the CPU parameter, it may be different from the time on the personal computer in use.

Error information

Display the error history registered to the diagnostic target module in the [Error Information] tab.

■Display content

The displayed information will differ depending on the module.

- CPU module: Up to 16 current errors will be displayed in the chronological order. The 17th and subsequence errors are not displayed.
- Other than CPU module: Up to 16 current errors are displayed.

If the same error as already displayed error code occurred, the display content is not updated.
For the errors of Q series module, occurrence date is not displayed.

■Error jump

Click the [Error Jump] button to jump to the selected error item.

Programs, function blocks, and each parameter are applicable for the Jump function.

A jump to parameters is available for the items supported by the Error jump function. The supported items differ depending on the module.

For inline structured text on ladder editor, the cursor will jump to the head line of the inline structured text regardless of the line that error occurs.

Precautions

In the following conditions, an error may not be found at the error jump destination.

- When the open project does not match with the data on the CPU module.
- When diagnosing programmable controller other than the one set as a connection destination in the project.

Ex.

When selecting a module from the "CC-Link IE Control Diagnostics" screen and activating the System Monitor function, and activating the "Module Diagnostics" screen from the activated "System Monitor" screen.

■Event history

☞ Page 322 Checking Error History/Operation History

■Clearing errors

After the errors displayed on the "Module Diagnostics" screen are cleared, click the [Clear Error] button to clear the error status of the module. (The LED that indicates error status on the front of the module turns OFF.)

The error content are also deleted from the list of the error information.

Module information list

The current LED information and switch information of the target module are displayed in the [Module Information List] tab.

16.3 Diagnosing Network

This section explains how to diagnose networks.

Ethernet diagnostics

Check the status of each connection, status of each protocol, and connection status.

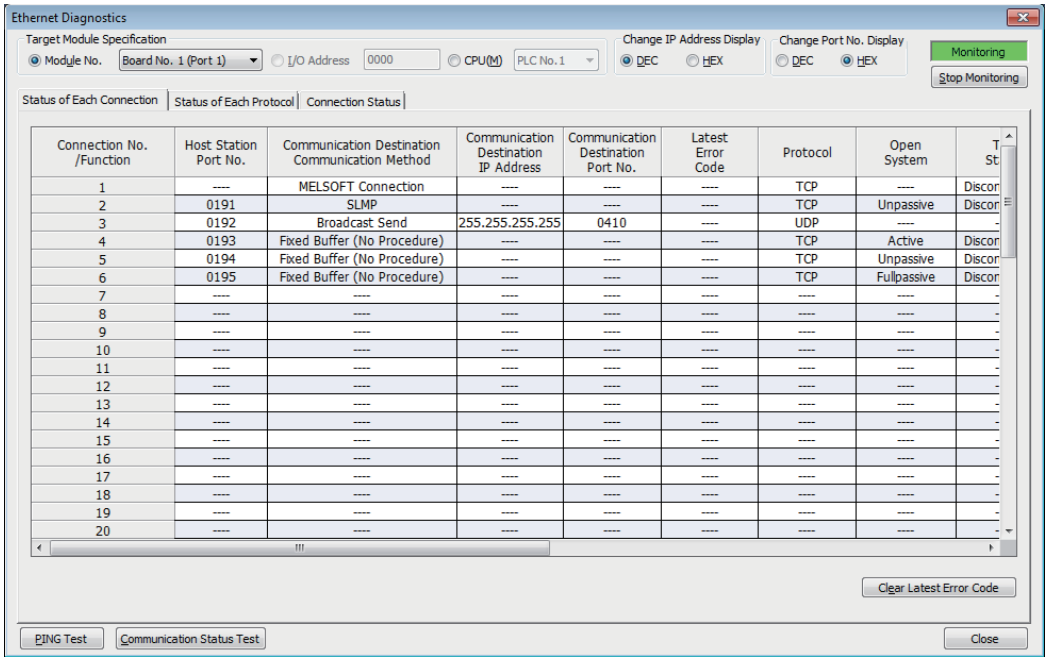
For details, refer to the following manuals.

📖 MELSEC iQ-R Ethernet User's Manual (Application)

📖 MELSEC iQ-F FX5 User's Manual (Ethernet Communication)

Window

[Diagnostics] ⇨ [Ethernet Diagnostics]



CC-Link IE Controller Network diagnostics

Monitor the network information, diagnose the network status, and perform a test.

FX5CPU does not support it.

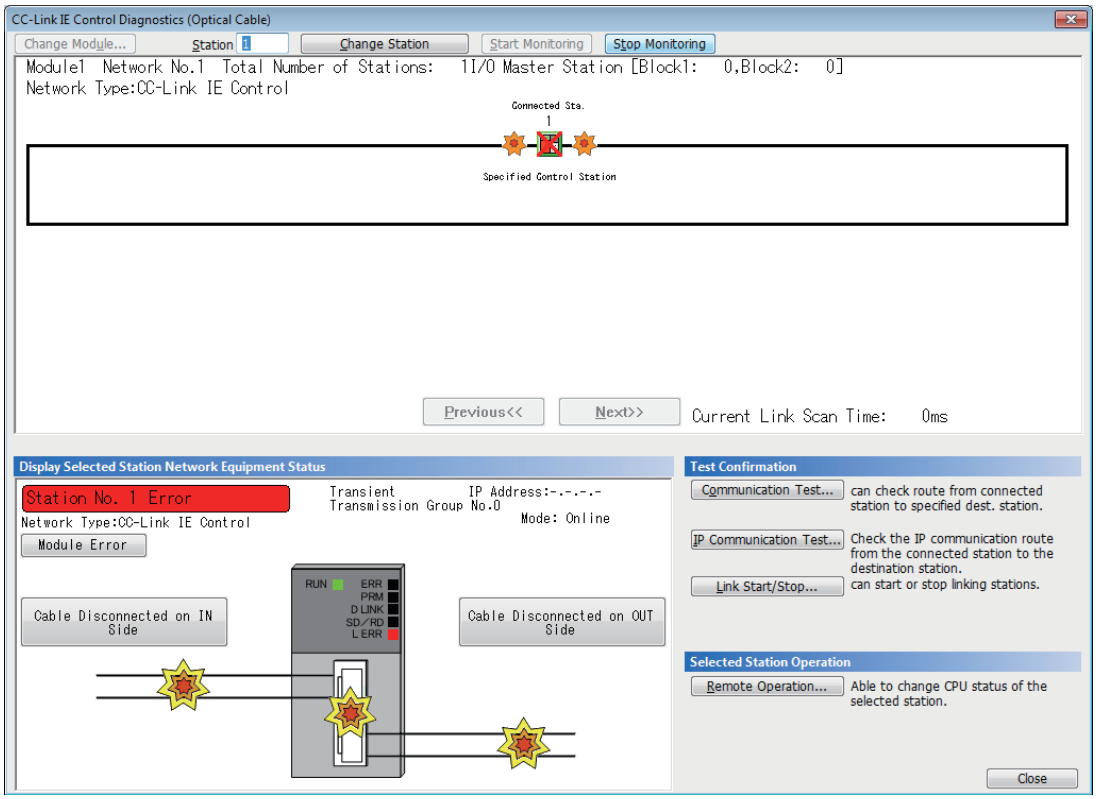
For details, refer to the following manual.

MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)

Window

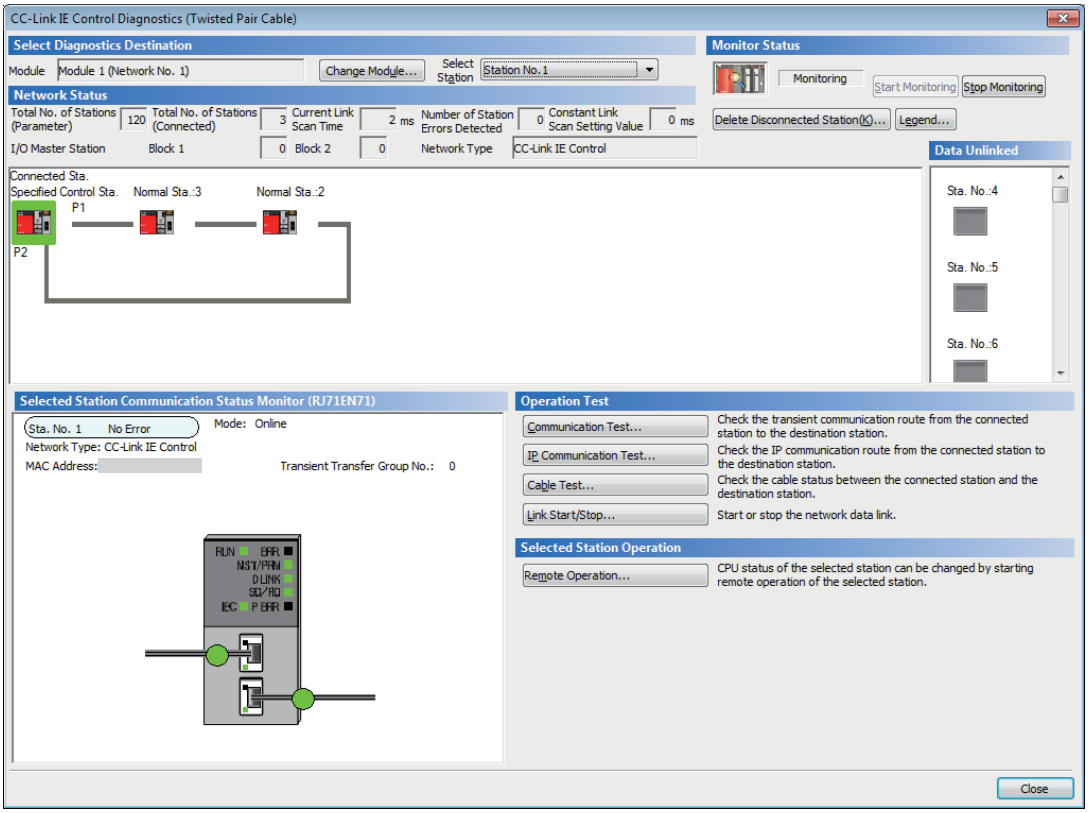
■Optical cable

[Diagnostics] ⇨ [CC-Link IE Control Diagnostics (Optical Cable)]



■Twisted pair cable

[Diagnostics] ⇨ [CC-Link IE Control Diagnostics (Twisted Pair Cable)]



CC-Link IE Field Network diagnostics

Monitor the network information, diagnose the network status, and perform a test.

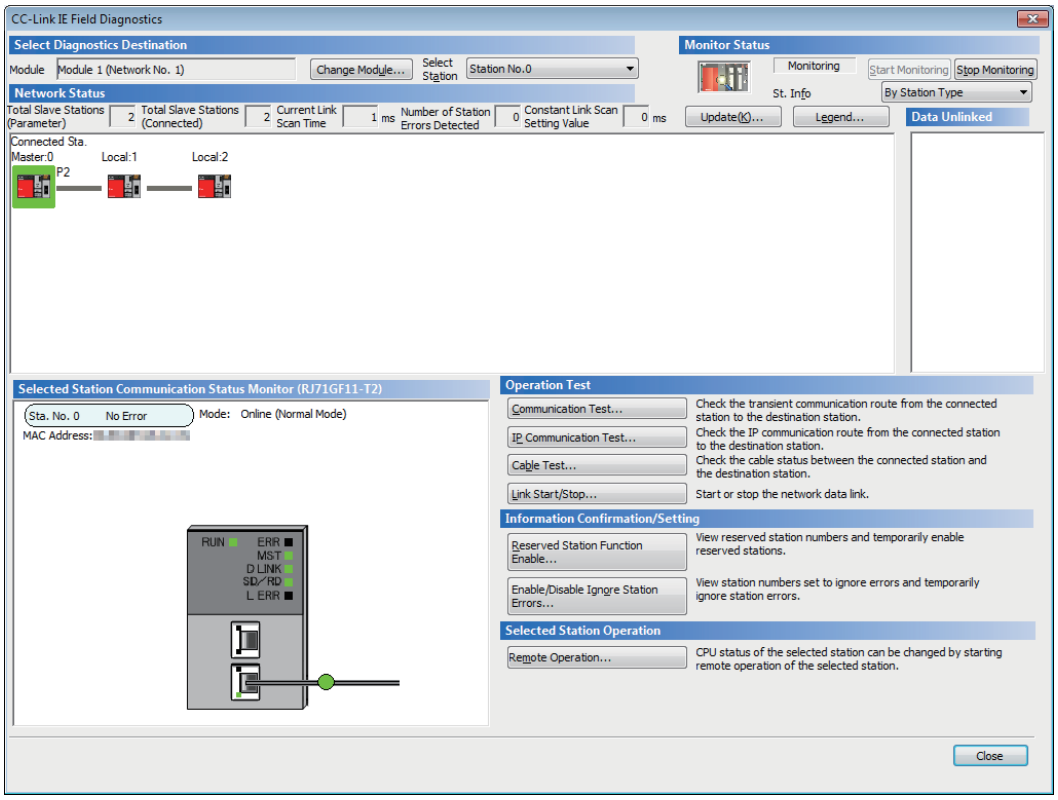
FX5CPU does not support it.

For details, refer to the following manual.

MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

Window

[Diagnostics] ⇨ [CC-Link IE Field Diagnostics]



MELSECNET diagnostics

Monitor the network information for each station, diagnose the network status, and perform a test.

RnPCPU/FX5CPU do not support this function.

Considerations for using MELSECNET/H network module when performing the MELSECNET Diagnostics , refer to the following manual.

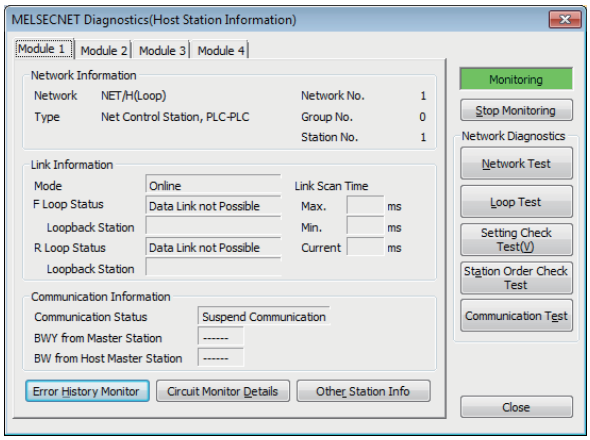
📖 MELSEC iQ-R Module Configuration Manual

For the operation method, refer to the following manual.

📖 GX Works2 Version 1 Operating Manual (Common)

Window

[Diagnostics] ⇨ [MELSECNET Diagnostics]



CC-Link diagnostics

Monitor the network information of each station, diagnose the network status, and perform a test.

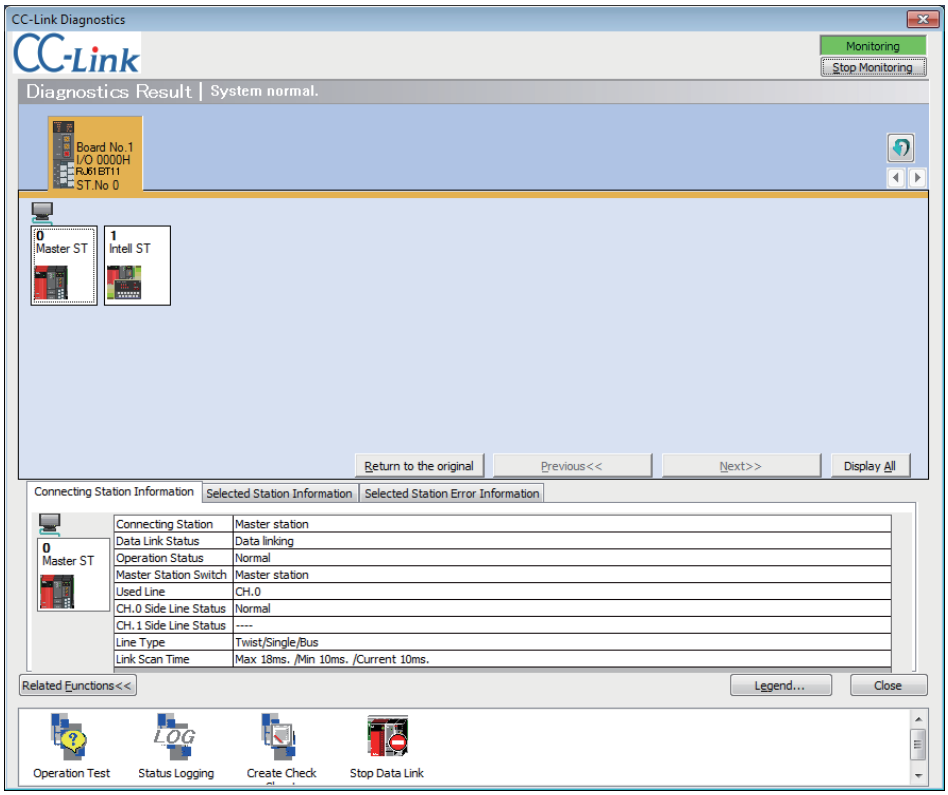
FX5CPU does not support it.

For details, refer to the following manual.

📖 MELSEC iQ-R CC-Link System Master/Local Module User's Manual (Application)

Window

[Diagnostics] ⇨ [CC-Link Diagnostics]



16.4 Checking Error History/Operation History

The module error information, operation history, and history of system information can be displayed using the Event History function.
The detailed information of the error history can be displayed when a CPU module and Intelligent function modules that support the Module error history collection function are used.
For the versions of modules that support this function, refer to the user's manual of each module.

Operating procedure

Click the [Event History] button on the "Module Diagnostics" screen or "System Monitor" screen.

Precautions

- Since the occurrence date for the Event History function depends on the time zone selected in "Clock Related Setting" for "Operation Related Setting" of the CPU parameter, it may be different from the time on the personal computer in use.
- When writing/reading data to/from the CPU module set the remote password, the event that the remote password lock/unlock is succeeded is registered more than once.

17 SAMPLING DEVICE DATA

For MELSEC-iQ-R series CPU modules, data can be sampled by using the data logging function.

17.1 Data Logging Function

This section explains the label and device data logging methods.

How to use the data logging function

1. Set the internal buffer capacity of the CPU module, which is used for the data logging function. (☞ Page 323 Setting internal buffer capacity for the data logging function)
2. Configure the data logging settings with CPU Module Logging Configuration Tool, and write the setting to the CPU module.
3. Start data logging with CPU Module Logging Configuration Tool.
4. Check the data logging results with GX LogViewer.

For information on how to use CPU Module Logging Configuration Tool, refer to the following manual.

📖 MELSEC iQ-R CPU Module User's Manual (Application)

For information on how to use GX LogViewer, refer to the following manual.

📖 GX LogViewer Version 1 Operating Manual

Setting internal buffer capacity for the data logging function

Set the internal buffer capacity for the data logging function using "Internal Buffer Capacity Setting" in "Memory/Device Setting" of the CPU parameter. Write the parameter to the CPU module after the setting is configured.

However, set the capacity within the range that meets the following conditions when the data logging function is used with the Memory Dump function and the realtime monitor function of CPU module.

The total of data logging buffer capacities + Memory dump buffer capacity + Realtime monitor buffer capacity ≤ 3072 KB

Each buffer capacity can be checked with the following:

- Data logging buffer capacity, Memory dump buffer capacity: CPU parameter
- Realtime monitor buffer capacity: The realtime monitor setting of GX LogViewer

MEMO

18 CHECKING/CHANGING CPU MODULE OPERATION

Check the operation status and change the operation of CPU module with GX Works3.

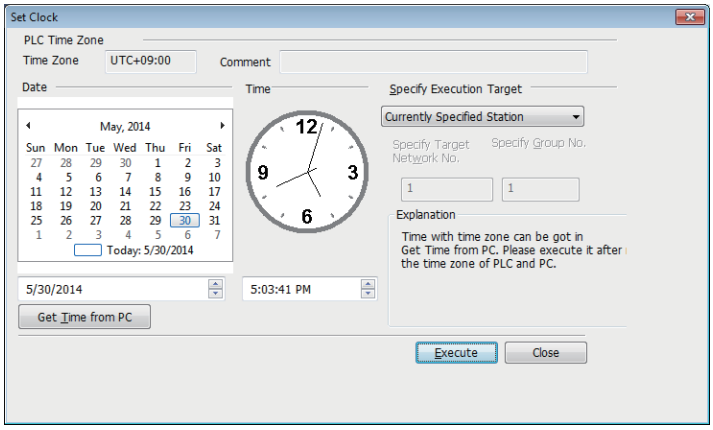
Purpose	Function name	Reference
Set the clock on CPU module.	Clock setting	Page 325 Setting Clock in CPU Module
RUN/STOP/PAUSE/RESET the CPU module connected to GX Works3.	Remote operation	Page 326 CPU Module Remote Operation
To perform the functions of RnSFCPU.	Switch Safety Operation Mode Check Safety Data Identity	Page 327 Operations of RnSFCPU
Check the usage of data memory.	Memory management	Page 330 Checking Memory Usage
Clear the values of data memory.	Clear value	Page 331 Initializing/Clearing Memory

18.1 Setting Clock in CPU Module

This section explains how to set the clock on a CPU module.

Window

[Online] ⇨ [Set Clock]



Operating procedure

Set the items and click the [Execute] button.

Execution target	Description
Currently Specified Station	Select this to set the clock only upon the station specified in the connection destination setting.
All Stations Specified*1	Select this to perform the clock setting on the station specified in the connection destination setting and all the stations on the same network. Set a target network with "Specify Network No.".
Specify Group No.*1	Select this to perform the clock setting on the station specified in the connection destination setting and each station of the specific group on the specified network. Set a target network in "Specify Target Network No.", and set a group number in "Specify Group No.". The clock setting cannot be performed for CC-Link IE Field Network because it does not have group number.

*1 Not supported by FX5CPU.

Precautions

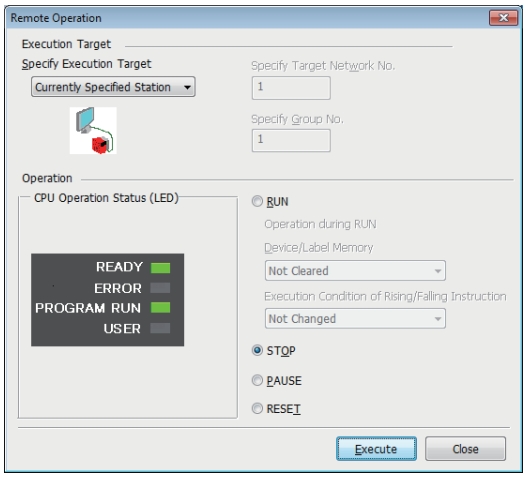
- The clock setting can be performed regardless of the ON/OFF state of the clock setting device 'SM210'. Note that the ON/OFF status of 'SM210' does not change upon clock setting.
- In the clock setting, a time-lag may occur due to the time of transfer.
- When different time zones are specified between the connected station and other station, the same time in the CPU is set to the target CPU module.

18.2 CPU Module Remote Operation

This section explains the switching method of the execution status of a CPU module (such as RUN/STOP) from GX Works3.

Window

[Online] ⇨ [Remote Operation]



Operating procedure

Set the items and click the [Execute] button.

Execution target	Description
Currently Specified Station	Select this to perform the remote operation only on the station specified in the connection destination setting.
All Stations Specified *1	Select this to perform the remote operation on the station specified in the connection destination setting and all the stations on the same network. Set a target network in "Specify Target Network No.".
Specify Group No. *1	Select this to perform the remote operation on the station specified in the connection destination setting and each station of the specific group on the specified network. Set a target network in "Specify Target Network No.", and set a group number in "Specify Group No.". The remote operation cannot be performed for CC-Link IE Field Network because it does not have group number.

*1 Not supported by FX5CPU.

Remote operation and RUN/STOP switch

When the operation to the CPU module differs by the remote operation, RUN/STOP switch, or remote RUN/PAUSE contact, the CPU module operates according to the priority shown below:

Operation to CPU module	Priority
STOP	1
PAUSE	2
RUN	3

The operating status after RUN/STOP switching on the CPU module or remote RUN/STOP contact operation are as follows.

Remote operation	CPU module switch		Remote RUN/STOP contact is ON (The CPU module is in STOP.)
	RUN	STOP	
RUN	RUN	STOP	STOP
STOP	STOP	STOP	STOP
PAUSE	PAUSE	STOP	STOP
RESET*1	Inoperable *2	RESET	RESET

*1 The remote reset operation must be enabled on "Operation Related Setting" of the CPU Parameter.
*2 Operable when the CPU module is switched to STOP by remote operation.

18.3 Operations of RnSFCPU

This section explains how to operate the functions of RnSFCPU with GX Works3.

Switch Safety Operation Mode

Switch the safety operation mode (safety mode/test mode) of RnSFCPU.

Considerations before switching the safety operation mode

⚠CAUTION

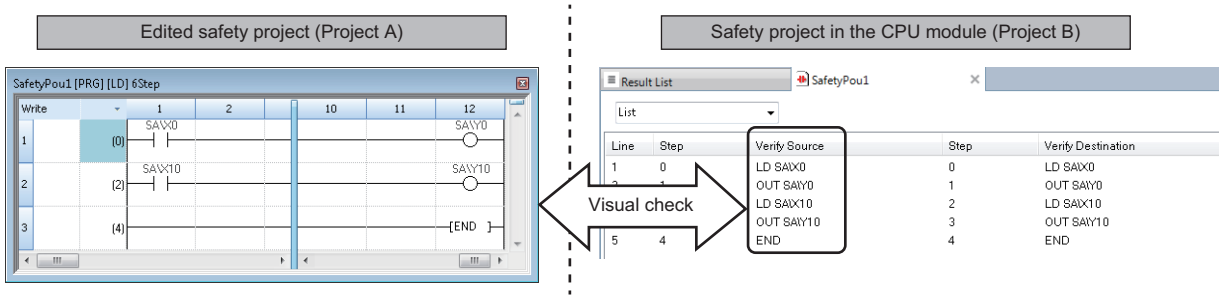
■Test the programs fully in the application environment before switching the mode to the safety mode.
■Before switching to the safety mode, ensure that the programs and parameters of the intended safety project have been written properly by the following operation.

Operating procedure

1. Save an edited safety project in the personal computer. (Hereinafter called 'Project A')
2. Start GX Works3, and write the programs and parameters of a safety project to CPU module. (☞ Page 250 Writing data to programmable controller)
3. Test the programs fully in the application environment.
4. Start another GX Works3, and connect it to RnSFCPU to read the programs and the parameters. (☞ Page 254 Reading data from programmable controller) (Hereinafter called 'Project B')
5. Visually check that there is no difference between Project A and Project B. (☞ Page 327 Checking programs, Page 328 Checking module parameters (Safety Communication Setting), Page 328 Checking parameters of safety remote I/O (slave station of CC-Link IE Field Network))

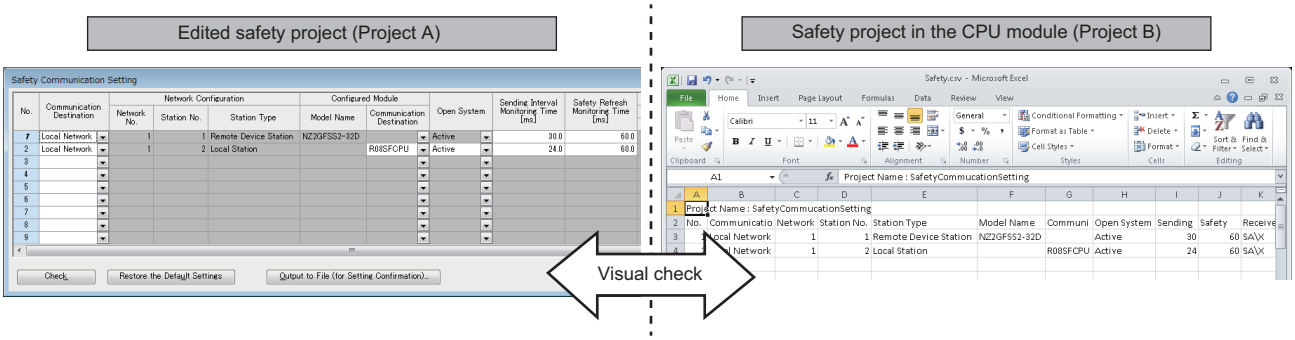
■Checking programs

Select Project A as the verify destination at Project B. Compare "Verify Source" on the detailed display of the "Verify Result" screen with the program editor of Project A to confirm whether the intended program has been written properly.



■Checking module parameters (Safety Communication Setting)

Output the "Safety Communication Setting" of module parameters in the Project B to a CSV file. Compare the saved CSV file with the "Safety Communication Setting" screen of the Project A to confirm whether the intended parameters have been written properly.

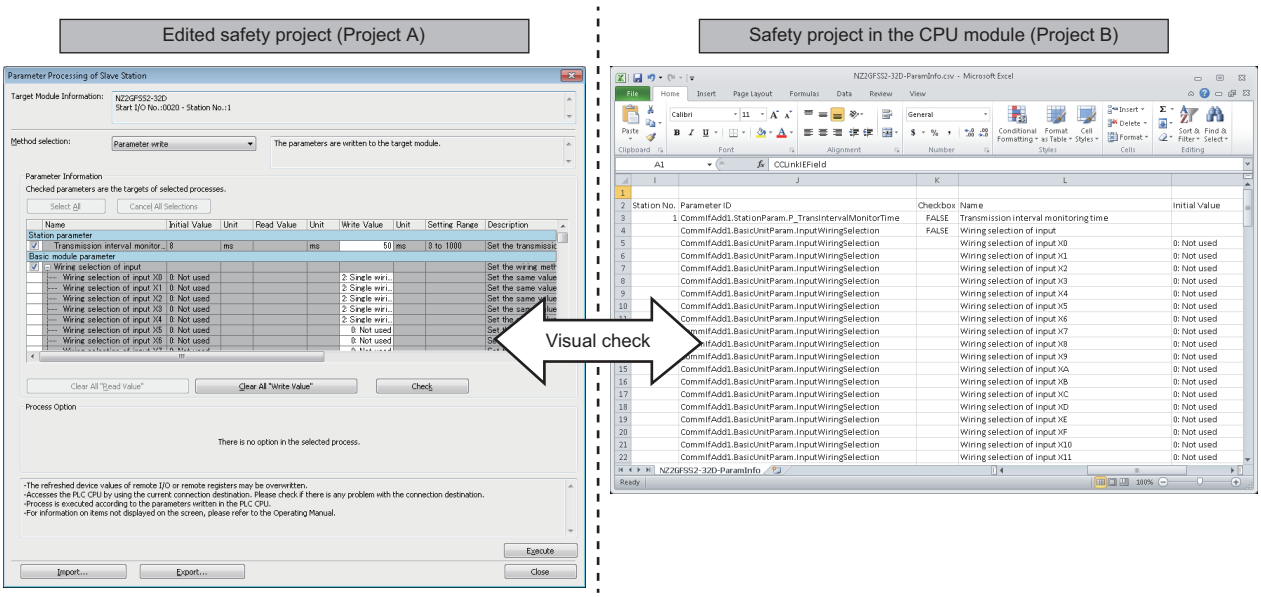


The figure above is a screen example in which Excel® is used. CSV files can also be checked by using other text editors.

■Checking parameters of safety remote I/O (slave station of CC-Link IE Field Network)

Check the following for all of the safety remote I/O modules in Project B.

Read the parameters from the slave station for CC-Link IE Field Network of Project B to a CSV file. Compare the saved CSV file with the "Parameter Processing of Slave Station" screen of Project A to confirm whether the intended parameters have been written properly.

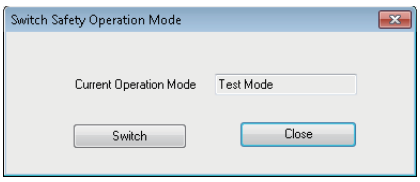


The figure above is a screen example in which Excel® is used. CSV files can also be checked by using other text editors.

Switching the safety operation mode

Window

[Online] ⇨ [Safety PLC Operation] ⇨ [Switch Safety Operation Mode]



Point

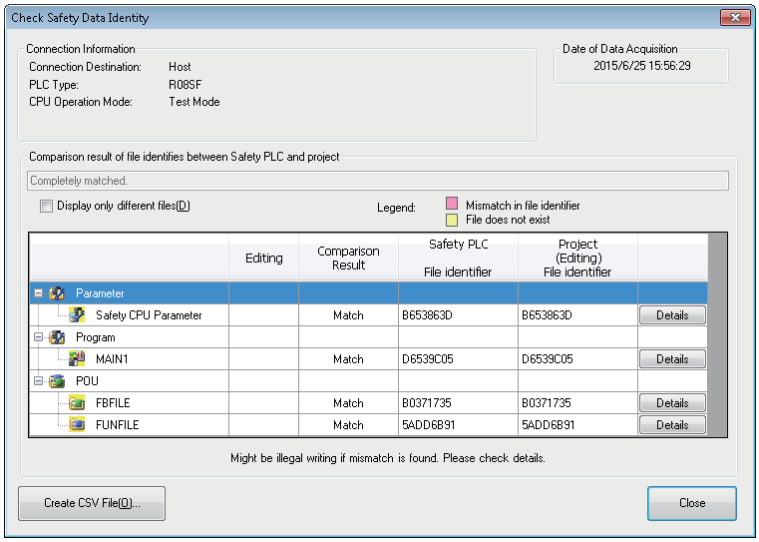
The safety operation mode of a running RnSFCPU can be checked in the [Module Information List] tab on the "Module Diagnostics" screen.

Check Safety Data Identity

Check whether the programs to be used in the safety mode and the project written to the CPU module are consistent.

Window

[Online] ⇨ [Safety PLC Operation] ⇨ [Check Safety Data Identity]



Checking whether the data in the project being open and in RnSFCPU are consistent

Check the result of compared data.

Checking whether the CPU module is running with the intended data

1. Save the information of Check Safety Data Identity in the CPU module as a CSV file using GX Works3 when the mode is changed to the safety mode after writing programs/parameters to the CPU module.
2. At a later date, perform the Check Safety Data Identity function with GX Works3, and check whether the "File identifier" in the saved CSV file and the one in the result are consistent.

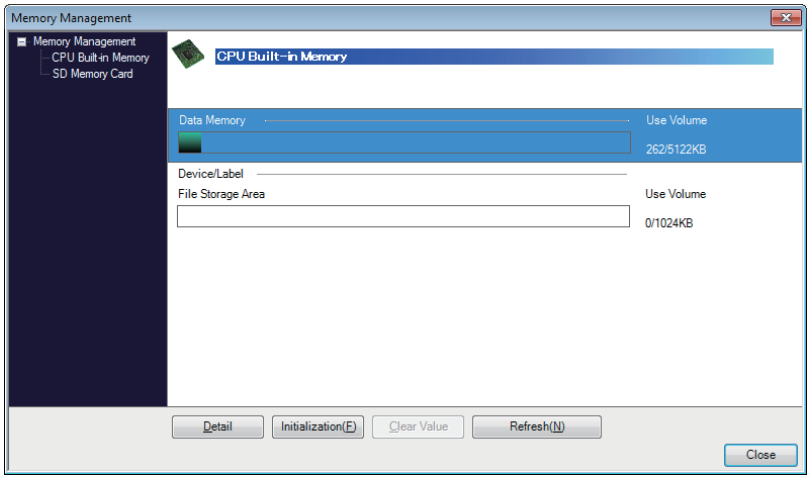
If the data are not consistent, the data in the CPU module may not be the one written by the user. In this case, do not use the data in the actual operation.

18.4 Checking Memory Usage

This section explains how to check the usage of data memory of CPU built-in memory, device/label memory, and SD memory card.

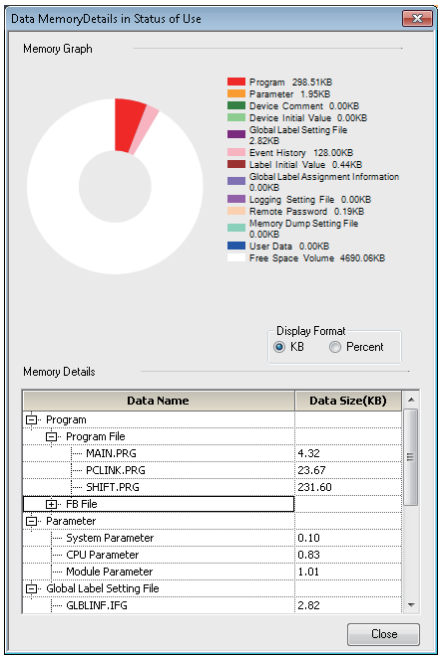
Window

[Online] ⇨ [CPU Memory Operation]



Details of memory usage

The details of current data memory, device/label memory, and data in an SD memory card can be checked by clicking the [Detail] button.



18.5 Initializing/Clearing Memory

This section explains how to initialize or clear the memory.
For details on the initialization and clearing to zero, refer to the following manuals.
📖 MELSEC iQ-R CPU Module User's Manual (Application)
📖 MELSEC iQ-F FX5 User's Manual (Application)

Initializing data

Delete files in the data memory/device memory/label memory of the CPU module, or folders/files stored in the SD memory card.
Note that the information for the user authentication will not be deleted. To delete the information, initialize all information.
(☞ Page 306 When a user name and a password used in the User Authentication function are lost)
After initializing, rewrite the programmable controller data to the CPU module.
If the power of the CPU is turned OFF → ON or the CPU module was reset, the allocation for the device/label memory is back to the initial status. (Each area has the default capacity.)

Operating procedure

1. Select [Online] ⇨ [CPU Memory Operation]. (☞ Page 330 Checking Memory Usage)
2. Select "Data Memory" or "File Storage Area" on the "Memory Management" screen, and click the [Initialization] button.

When initializing an SD memory card, select "SD Memory Card" displayed on the left of the screen.

Precautions

Format the memory card to be used for a CPU module by using this function.
If a memory card is formatted using the function such as Windows® format function, it may not be used on the CPU module.

Clearing value

Clear the values of devices, labels, and file registers to '0', and perform latch clear operation.

Operating procedure

1. Select [Online] ⇨ [CPU Memory Operation]. (☞ Page 330 Checking Memory Usage)
2. Select "Device/Label Memory" on the "Memory Management" screen, and click the [Clear Value] button.
3. Set the items on the "Clear Value" screen, and click the [Execute] button.

Point

When the device initial value file/label initial value file is written in the CPU module, it can be initialized by resetting the CPU module.

APPENDIX

Appendix 1 Functions Added to/Changed from Previous Version

The table below shows the main functions added/changed along with the upgrades.

Supported version	Main functions	Added/changed contents	Reference
Version 1.001B	—	• The problems causing an error when Version 1.000A is used have been solved.	—
	Online program change	• Online program change on an instruction basis in ladder programs is supported. • Online program change for ST programs is supported. • Online program change after changing/deleting label initial values is supported.	Page 258 Writing Programs While CPU is in RUN
Version 1.005F	Switching displayed language	• Chinese Simplified is supported.	Page 27 Switching Displayed Language
	Checking program	• The Check Program function is supported.	Page 183 Checking Programs
	Online program change	• Performing/not performing the device comment difference check before online program change can be selected with option.	Page 258 Writing Programs While CPU is in RUN
	Printing data	• Printing the product information list is supported.	Page 51 Printing Data
	Opening GX Works2 format projects	• The automatic input of the start XY for I/O assignment is supported. • The basic settings for I/O assignment (base, power supply, cable) is succeeded. • The device memory and device initial value are succeeded.	Page 64 Opening GX Works2 format project
	Interaction with iQ Works*1	• The system label is supported.	Page 106 REGISTERING LABELS
		• The parameter reflection function of MELSOFT Navigator is supported.	—
	Memory dump*2	• The Memory Dump function is supported.	Page 284 Saving and Checking Device Value
	Module tool list	• "Offset/gain setting" under "Temperature Input" for "iQ-R Series" is supported. • "Q61LD Two-Point Calibration Setting" under "Analog Module" for "Q Series" is supported. • "Q61LD Default Setting" under "Analog Module" for "Q Series" is supported.	Page 102 Other settings of Intelligent function module
	Diagnostics*3	• The CC-Link IE Controller Network diagnostics (Twisted pair cable) is supported.	Page 317 CC-Link IE Controller Network diagnostics
Version 1.007H	Function Block Diagram/ Ladder Diagram language*2	• Function Block Diagram/Ladder Diagram language is supported.	Page 151 Creating FBD/LD Programs
	RnPCPU	• RnPCPU (R08PCPU, R16PCPU, R32PCPU, and R120PCPU) is supported.	—
	FX5CPU	• FX5CPU (FX5UCPU and FX5UCCPU) is supported.	—
	Simulation	• The Simulation function of RnCPU (R04CPU, R08CPU, R16CPU, R32CPU, and R120CPU) is supported.	Page 226 SIMULATING PROGRAM
	Opening GX Works2 format projects	• The PLC file setting and the PLC RAS setting of PLC parameter are succeeded. • Structured Ladder/FBD programs are succeeded.	Page 64 Opening GX Works2 format project
	Interaction with iQ Works	• A system label change notification can be received automatically.	Page 118 Importing the changes of system label database
	Module tool list	• "Analog input - Offset/gain setting" under "Analog Adapter" for "FX5 Series" is supported. • "Analog output - Offset/gain setting" under "Analog Adapter" for "FX5 Series" is supported.	Page 102 Other settings of Intelligent function module
	Automatic detection of CC-Link IE Field Network connected devices	• The connected devices on the CC-Link IE Field Network can be detected automatically.	—

Supported version	Main functions	Added/changed contents	Reference
Version 1.010L	Project Revision	• Project Revision function is supported.	Page 80 Project Revision
	Library	• User library is supported.	Page 219 Enhancing Use of User Library
	Q series MELSECNET/H network module	• Q series MELSECNET/H network module is supported. • MELSECNET Diagnostics is supported.	— Page 320 MELSECNET diagnostics
	Opening GX Works2 format projects	• The Built-in Ethernet Port Setting of PLC parameter is succeeded.	Page 64 Opening GX Works2 format project
	Verification	• Verifying labels is supported.	Page 77 Verifying Projects
	Printing data	Printing the following data is supported. • Device comment, device memory, device initial value • Statement/note • Device list • Cross reference • Verification result	Page 51 Printing Data
	Changing ladder block temporarily	• Ladder block can be changed temporarily.	Page 141 Changing ladder blocks temporarily
Version 1.015R	Sequential Function Chart*2	• Sequential Function Chart is supported.	Page 158 Creating SFC Programs
	RnENCPU	• RnENCPU (R04ENCPU, R08ENCPU, R16ENCPU, R32ENCPU, R120ENCPU) is supported.	—
	RnSFCPU	• RnSFCPU(R08SFCPU, R16SFCPU, R32SFCPU, R120SFCPU) is supported. • CPU extension modules (R6SFM) are supported.	—
	Execution order setting	• The execution order setting of progaram blocks is supported.	Page 121 Program execution order/execution type settings
	Simulation	• The I/O System Setting function is supported.	Page 229 External Device Operation Simulation
	Offline Monitor	• Device data collected with the memory dump function can be monitored on each program editor.	Page 288 Checking Sampled Data on Program Editor
	Opening GX Works2 format projects	• The Boot File Setting, SFC Setting, and Multiple CPU Setting of PLC parameter are succeeded.	Page 64 Opening GX Works2 format project
	Verification	• Differences of verification results can be checked in the editor format.	Page 78 Detailed display (editor format)
	Writing/reading data to/from a memory card	• Data can be written/read to/from a memory card inserted into a personal computer.	Page 264 Writing/Reading Data to/from Memory Card
	Module tool list	• Temperature control modules (R60TCTRT2TT2, R60TCTRT2TT2BW, R60TCRT4, R60RCRT4BW) are supported.	Page 102 Other settings of Intelligent function module

*1 To use this function, it is required that MELSOFT Navigator supports the function. For information on the version of the supported MELSOFT Navigator, refer to the Help of MELSOFT Navigator.

*2 To use this function, it is required that CPU module supports the function. For information on the firmware version of the supported CPU modules, refer to the following manual.
MELSEC iQ-R CPU Module User's Manual (Application)

*3 To use this function, it is required that the module supports the function. For information on the firmware version of the supported module, refer to the manual for respective modules.

Appendix 2 Unusable Character String (Reserved Word)

Character strings used for application function names, common instruction names, special instruction names, instructions and so on are referred to as the reserved words.
Reserved words may not be used for names.
When the character strings defined as reserved words is used for names, an error occurs when performing registration or conversion.

Precautions

Names are not case-sensitive.

Unusable character string for project name/work space name

Category		Character string
Invalid character	Symbol	Space, ", %, ', *, /, ., :, <, >, ?, \, |, 0x007f
	Surrogate pair	0xD800 to 0xDBFF, 0xDC00 to 0xDFFF
	Control code	U+0000 to U+001F, U+0080 to U+009F, U+00A0 to U+00BF, U+FFFE, U+FFFF
Windows® reserved word		COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, AUX, CON, PRN, NUL, CLOCK$, END_MARK

Unusable character string for data name

Category		Character string
Invalid character	Symbol*1	Space, !, ", #, $, %, &, ', (,), *, +, /, ,, -, ., :, ;, <, =, >, ?, @, [, \,], ^, `, {, |, }, ~, 0x007f
	Surrogate pair	0xD800 to 0xDBFF, 0xDC00 to 0xDFFF
	Control code	U+0000 to U+001F, U+0080 to U+009F, U+00A0 to U+00BF, U+FFFE, U+FFFF
	Others*1	COMMENT, GLBLINF
Windows® reserved word		COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, AUX, CON, PRN, NUL, CLOCK$, END_MARK
Data type reserved words	Class*1	VAR, VAR_RETAIN, VAR_ACCESS, VAR_CONSTANT, VAR_INPUT, VAR_INPUT_RETAIN, VAR_OUTPUT, VAR_OUTPUT_RETAIN, VAR_IN_OUT, VAR_IN_EXT, VAR_EXTERNAL, VAR_EXTERNAL_CONSTANT, VAR_EXTERNAL_RETAIN, VAR_GLOBAL, VAR_GLOBAL_CONSTANT, VAR_GLOBAL_RETAIN, VAR_PUBLIC, VAR_PUBLIC_RETAIN
	Data type	BOOL, BYTE, INT, SINT, DINT, LINT, UINT, USINT, UDINT, ULINT, WORD, DWORD, LWORD, ARRAY, REAL, LREAL, TIME, STRING, WSTRING, TIMER, LTIMER, RETENTIVETIMER, LRETENTIVETIMER, COUNTER, LCOUNTER, POINTER ANY, ANY_NUM, ANY_BIT, ANY_REAL, ANY_INT, ANY_DATE, ANY_SIMPLE, ANY16, ANY32, DATE, DATE_AND_TIME, DT, TIME, TIME_OF_DAY, TOD, BODY_CCE, BODY_FBD, BODY_IL, BODY_LD, BODY_SFC, BODY_ST, END_BODY, END_PARAMETER_SECTION, PARAM_FILE_PATH, PARAMETER_SECTION, POW, LONGCOUNTER, LONGRETENTIVETIMER, LONGTIMER, UNKNOWN, ANY_BITADDR, ANY_WORDADDR, ANY_STRING, ANYSTRING_SINGLE, ANYSTRING_DOUBLE, ANY_ELEMENTARY, ANY_ELEMENTARY_IEC, ANY_MAGNITUDE, ANY_BOOL, ANY_SIGNED, ANY_UNSIGNED, ANYREAL_64, ANYREAL_32, ANY_DERIVED, ANY16_S, ANY16_U, ANY32_S, ANY32_U, ANY64, ANY64_S, ANY64_U, ANY_STRUCT, ANYWORD_ARRAY, ANY16_ARRAY, ANY16_S_ARRAY, ANY16_U_ARRAY, ANY32_ARRAY, ANY32_S_ARRAY, ANY32_U_ARRAY, ANY64_ARRAY, ANY64_S_ARRAY, ANY64_U_ARRAY, ANY_REAL_ARRAY, ANY_REAL_32_ARRAY, ANY_REAL_64_ARRAY, ANY_STRING_ARRAY, ANYSTRING_SINGLE_ARRAY, ANYSTRING_DOUBLE_ARRAY, ANYBIT_ARRAY, UINT_WORD, UDINT_DWORD, ULINT_LWORD, TIME2, KBIT64, KBIT32, KBIT16, WDEVICE, BDEVICE, DUMMYDEVICE, FB, FUN, STRUCT_MEMBER, BIT_ARRAY, STRUCT_ARRAY, STRING_ARRAY, WSTRING_ARRAY, LINT_ARRAY, ULINT_LWORD_ARRAY, DINT_ARRAY, UDINT_DWORD_ARRAY, INT_ARRAY, UINT_WORD_ARRAY, REAL_ARRAY, LREAL_ARRAY, TIME_ARRAY, NONE, ANY16_OR_STRING_SINGLE, ANY_DT, ANY_TM, ANY_BOOL_OR_POINTER

Category		Character string
IEC reserved word*1		ABS, ACOS, ACTION, ADD, AND, ANDN, ANY, ANY_BIT, ANY_INT, ANY_NUM, ANY_REAL, ARRAY, ASIN, AT, ATAN, BOOL, BY, BYTE, CAL, CALC, CASE, CONCAT, CONFIGURATION, CONSTANT, COS, CTD, CTU, CTUD, D, DATE, DATE_AND_TIME, DELETE, DINT, DIV, DO, DT, DWORD, ELSE, ELSIF, END_ACTION, END_CASE, END_CONFIGURATION, END_FOR, END_FUNCTION, END_FUNCTION_BLOCK , END_IF, END_PROGRAM, END_REPEAT, END_RESOURCE, END_STEP, END_STRUCT, END_TRANSITION, END_TYPE, END_VAR, END_WHILE, EN, ENO, EQ, EXIT, EXP, EXPT, FALSE, F_EDGE, F_TRIG, FIND, FOR, FROM, FUNCTION, FUNCTION_BLOCK, GE, GT, IF, INITIAL_STEP, INSERT, INT, INTERVAL, JMP, JMPC, JMPCN, L, LD, LDN, LE, LEFT, LEN, LIMIT, LINT, LN, LOG, LREAL, LT, LWORD, MAX, MID, MIN, MOD, MOVE, MUL, MUX, N, NE, NEG, NOT, OF, ON, OR, ORN, P, PROGRAM, R, R1, R_TRIG, READ_ONLY, READ_WRITE, REAL, REPEAT, REPLACE, RESOURCE, RET, RETAIN, RETC, RETURN, RIGHT, ROL, ROR, RS, R_EDGE, S, S1, SD, SEL, SHL, SHR, SIN, SINGLE, SINT, SQRT, SR, ST, STEP, STN, STRING, STRUCT, SUB, TAN, TASK, THEN, TIME, TIME_OF_DAY, TO, TOD, TOF, TON, TP, TRANSITION, TRUE, TYPE, UDINT, UINT, ULINT, UNTIL, USINT, VAR, VAR_ACCESS, VAR_EXTERNAL, VAR_GLOBAL, VAR_INPUT, VAR_IN_OUT, VAR_OUTPUT, WHILE, WITH, WORD, XOR, XORN
Code conversion*1	Device	A, B, BL, C, CC, CN, CS, D, DX, DY, E, F, FD, FX, FY, GLP, H, I, J, K, L, LC, LCC, LCN, LCS, LLP, LSC, LSN, LSS, LST, LSTC, LSTN, LSTS, LT, LTC, LTN, LTS, LZ, M, N, P, PH, R, RD, S, SB, SC, SD, SM, SN, SS, ST, STC, STN, STS, SW, SZ, T, TC, TN, TR, TS, U, V, VD, W, X, Y, Z, ZR, ZZ
	Others	Maker-defined FB/FUN, MELSEC instructions

*1 These can be used for some names.

Unusable character string for label name

Category		Character string
Invalid character	Symbol	Space, !, ", #, $, %, &, ', (,), *, +, /, ,, -, ., :, ;, <, =, >, ?, @, [, \,], ^, `, {, |, }, ~, 0x007f
	Surrogate pair	0xD800 to 0xDBFF, 0xDC00 to 0xDFFF
	Control code	U+0000 to U+001F, U+0080 to U+009F, U+00A0 to U+00BF, U+FFFE, U+FFFF
Windows® reserved word		COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, AUX, CON, PRN, NUL, CLOCK$, END_MARK
Data type reserved words	Class	VAR, VAR_RETAIN, VAR_ACCESS, VAR_CONSTANT, VAR_INPUT, VAR_INPUT_RETAIN, VAR_OUTPUT, VAR_OUTPUT_RETAIN, VAR_IN_OUT, VAR_IN_EXT, VAR_EXTERNAL, VAR_EXTERNAL_CONSTANT, VAR_EXTERNAL_RETAIN, VAR_GLOBAL, VAR_GLOBAL_CONSTANT, VAR_GLOBAL_RETAIN, VAR_PUBLIC, VAR_PUBLIC_RETAIN
	Data type	BOOL, BYTE, INT, SINT, DINT, LINT, UINT, USINT, UDINT, ULINT, WORD, DWORD, LWORD, ARRAY, REAL, LREAL, TIME, STRING, WSTRING, TIMER, LTIMER, RETENTIVETIMER, LRETENTIVETIMER, COUNTER, LCOUNTER, POINTER ANY, ANY_NUM, ANY_BIT, ANY_REAL, ANY_INT, ANY_DATE, ANY_SIMPLE, ANY16, ANY32, DATE, DATE_AND_TIME, DT, TIME, TIME_OF_DAY, TOD, BODY_CCE, BODY_FBD, BODY_IL, BODY_LD, BODY_SFC, BODY_ST, END_BODY, END_PARAMETER_SECTION, PARAM_FILE_PATH, PARAMETER_SECTION, POW, LONGCOUNTER, LONGRETENTIVETIMER, LONGTIMER, UNKNOWN, ANY_BITADDR, ANY_WORDADDR, ANY_STRING, ANYSTRING_SINGLE, ANYSTRING_DOUBLE, ANY_ELEMENTARY, ANY_ELEMENTARY_IEC, ANY_MAGNITUDE, ANY_BOOL, ANY_SIGNED, ANY_UNSIGNED, ANYREAL_64, ANYREAL_32, ANY_DERIVED, ANY16_S, ANY16_U, ANY32_S, ANY32_U, ANY64, ANY64_S, ANY64_U, ANY_STRUCT, ANYWORD_ARRAY, ANY16_ARRAY, ANY16_S_ARRAY, ANY16_U_ARRAY, ANY32_ARRAY, ANY32_S_ARRAY, ANY32_U_ARRAY, ANY64_ARRAY, ANY64_S_ARRAY, ANY64_U_ARRAY, ANY_REAL_ARRAY, ANY_REAL_32_ARRAY, ANY_REAL_64_ARRAY, ANY_STRING_ARRAY, ANYSTRING_SINGLE_ARRAY, ANYSTRING_DOUBLE_ARRAY, ANYBIT_ARRAY, UINT_WORD, UDINT_DWORD, ULINT_LWORD, TIME2, KBIT64, KBIT32, KBIT16, WDEVICE, BDEVICE, DUMMYDEVICE, FB, FUN, STRUCT_MEMBER, BIT_ARRAY, STRUCT_ARRAY, STRING_ARRAY, WSTRING_ARRAY, LINT_ARRAY, ULINT_LWORD_ARRAY, DINT_ARRAY, UDINT_DWORD_ARRAY, INT_ARRAY, UINT_WORD_ARRAY, REAL_ARRAY, LREAL_ARRAY, TIME_ARRAY, NONE, ANY16_OR_STRING_SINGLE, ANY_DT, ANY_TM, ANY_BOOL_OR_POINTER
IEC reserved word		ABS, ACOS, ACTION, ADD, AND, ANDN, ANY, ANY_BIT, ANY_INT, ANY_NUM, ANY_REAL, ARRAY, ASIN, AT, ATAN, BOOL, BY, BYTE, CAL, CALC, CASE, CONCAT, CONFIGURATION, CONSTANT, COS, CTD, CTU, CTUD, D, DATE, DATE_AND_TIME, DELETE, DINT, DIV, DO, DT, DWORD, ELSE, ELSIF, END_ACTION, END_CASE, END_CONFIGURATION, END_FOR, END_FUNCTION, END_FUNCTION_BLOCK , END_IF, END_PROGRAM, END_REPEAT, END_RESOURCE, END_STEP, END_STRUCT, END_TRANSITION, END_TYPE, END_VAR, END_WHILE, EN, ENO, EQ, EXIT, EXP, EXPT, FALSE, F_EDGE, F_TRIG, FIND, FOR, FROM, FUNCTION, FUNCTION_BLOCK, GE, GT, IF, INITIAL_STEP, INSERT, INT, INTERVAL, JMP, JMPC, JMPCN, L, LD, LDN, LE, LEFT, LEN, LIMIT, LINT, LN, LOG, LREAL, LT, LWORD, MAX, MID, MIN, MOD, MOVE, MUL, MUX, N, NE, NEG, NOT, OF, ON, OR, ORN, P, PROGRAM, R, R1, R_TRIG, READ_ONLY, READ_WRITE, REAL, REPEAT, REPLACE, RESOURCE, RET, RETAIN, RETC, RETURN, RIGHT, ROL, ROR, RS, R_EDGE, S, S1, SD, SEL, SHL, SHR, SIN, SINGLE, SINT, SQRT, SR, ST, STEP, STN, STRING, STRUCT, SUB, TAN, TASK, THEN, TIME, TIME_OF_DAY, TO, TOD, TOF, TON, TP, TRANSITION, TRUE, TYPE, UDINT, UINT, ULINT, UNTIL, USINT, VAR, VAR_ACCESS, VAR_EXTERNAL, VAR_GLOBAL, VAR_INPUT, VAR_IN_OUT, VAR_OUTPUT, WHILE, WITH, WORD, XOR, XORN
Code conversion	Device	A, B, BL, C, CC, CN, CS, D, DX, DY, E, F, FD, FX, FY, GLP, H, I, J, K, L, LC, LCC, LCN, LCS, LLP, LSC, LSN, LSS, LST, LSTC, LSTN, LSTS, LT, LTC, LTN, LTS, LZ, M, N, P, PH, R, RD, S, SB, SC, SD, SM, SN, SS, ST, STC, STN, STS, SW, SZ, T, TC, TN, TR, TS, U, V, VD, W, X, Y, Z, ZR, ZZ
	Others	Maker-defined FB/FUN, MELSEC instructions

Unusable character strings for path name

Category		Character string
Invalid character	Symbol	", *, /, ,, ;, <, >, ?, |, 0x007f
	Surrogate pair	0xD800 to 0xDBFF, 0xDC00 to 0xDFFF
	Control code	U+0000 to U+001F, U+0080 to U+009F, U+00A0 to U+00BF, U+FFFE, U+FFFF
Windows® reserved word		COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, AUX, CON, PRN, NUL, CLOCK$, END_MARK

Unusable character string for library name

Category		Character string
Invalid character	Symbol	Space, !, ", #, $, %, &, ', (,), *, +, /, ,, -, ., :, ;, <, =, >, ?, @, [, \,], ^, `, {, |, }, ~, 0x007f
	Surrogate pair	0xD800 to 0xDBFF, 0xDC00 to 0xDFFF
	Control code	U+0000 to U+001F, U+0080 to U+009F, U+00A0 to U+00BF, U+FFFE, U+FFFF
Windows® reserved word		COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, AUX, CON, PRN, NUL, CLOCK$, END_MARK

Appendix 3 Applicable Devices in GX Works3

For RCPU

○: Applicable, △: Display only, ×: Not applicable —: Not available

Category	Device name	Symbol	Digit specification	Bit-specified word device	Device comment	Device memory	Device initial value
User device	Input	X	○	—	○	△	×
	Output	Y	○	—	○	△	×
	Internal relay	M *1	○	—	○	○	×
	Latch relay	L	○	—	○	○	×
	Link relay	B	○	—	○	○	×
	Annunciator	F	○	—	○	○	×
	Link special relay	SB	○	—	○	△	×
	Edge relay	V*1	○	—	○	○	×
	Step relay	S*2	○	—	×	△	×
	Timer	T *1	—	×	○	○	○
	Retentive timer	ST *1	—	×	○	○	○
	Long timer	LT*1	—	×	○	○	○
	Retentive long timer	LST*1	—	×	○	○	○
	Counter	C *1	—	×	○	○	○
	Long counter	LC *1	—	×	○	○	○
	Data register	D *1	—	○	○	○	○
	Link register	W	—	○	○	○	○
	Link special register	SW	—	○	○	○	○
System device	Function input	FX	×	—	×	×	×
	Function output	FY	×	—	×	×	×
	Special relay	SM	○	—	○	△	×
	Function register	FD	—	○	×	×	×
	Special register	SD	—	○	○	○	○
Link direct device (J□\□)	Link input	X	○	—	○	△	×
	Link output	Y	○	—	○	△	×
	Link relay	B	○	—	○	△	×
	Link special relay	SB	○	—	○	△	×
	Link register	W	—	○	○	○	○
	Link special register	SW	—	○	○	○	○
Module access device (U□\G□)	Module access device	G	—	○	○	○	○
CPU buffer memory access device (U3E□\G□/HG□)	CPU buffer memory access device	G/HG	—	○	○	○	○
Index register	Index register	Z *1	—	×	×	○	×
	Long index register	LZ *1	—	×	×	○	×
File register	File register	R	—	○	○	×	×
		ZR	—	○	○	○	○
Module refresh register	Module refresh register	RD	—	○	○	○	○
Nesting	Nesting	N	—	—	×	×	×
Pointer	Pointer	P *1	—	—	○	×	×
	Interrupt pointer	I	—	—	○	×	×
Other devices	SFC block device	BL	—	—	○	×	×
	SFC transition device	TR	—	—	×	×	×
	Network number specified device	J	—	—	○	×	×
	I/O number specified device	U	—	—	○	×	×

Category	Device name	Symbol	Digit specification	Bit-specified word device	Device comment	Device memory	Device initial value
Constant	Decimal constant	K	—	—	×	×	×
	Hexadecimal constant	H	—	—	×	×	×
	Real constant	E	—	—	×	×	×
	String constant	—	—	—	×	×	×

*1 Writing to local devices is available.
*2 When editing it in other places except in a Zoom, specify a step relay with a block specification (BL□\S□).

How to input local devices

Prefix '#' to the device name (Example: #D10)

Creating comments for module access devices

Device comments within the following range can be created in the buffer memory (3E00H to 3E30H) of CPU module in a multiple CPU system.

Supported range	Supported range
U0 (\G0) to U1FF (\G268435455)	U200 (\G0) to U3DF (\G268435455)
U3E0 (\G0) to U3E3 (\G268435455)	

Applicable devices in safety project

○: Applicable, △: Display only, ×: Not applicable —: Not available

Category	Device name	Symbol	Digit specification	Bit-specified word device	Device comment	Device memory	Device initial value
User device	Safety input	SA\X	○	—	○	△	×
	Safety output	SA\Y	○	—	○	△	×
	Safety internal relay	SA\M*1	○	—	○	○	×
	Safety link relay	SA\B	○	—	○	○	×
	Safety timer	SA\T*1	—	×	○	○	×
	Safety retentive timer	SA\ST*1	—	×	○	○	×
	Safety counter	SA\C*1	—	×	○	○	×
	Safety data register	SA\D*1	—	○	○	○	×
	Safety link register	SA\W	—	○	○	○	×
System device	Safety special relay	SA\SM	○	—	○	△	×
	Safety special register	SA\SD	—	○	○	○	×

*1 The devices can be written to local devices.

■How to input device

Global device: Prefix 'SA\' to the device name. (Example: SA\D10)

Local device: Prefix 'SA\#' to the device name. (Example: SA\#D10)

For FX5CPU

○: Applicable, △: Display only, ×: Not applicable —: Not available

Category	Device name	Symbol	Digit specification	Bit-specified word device	Device comment	Device memory	Device initial value
User device	Input	X	○	—	○	△	×
	Output	Y	○	—	○	△	×
	Internal relay	M	○	—	○	○	×
	Latch relay	L	○	—	○	○	×
	Link relay	B	○	—	○	○	×
	Annunciator	F	○	—	○	○	×
	Link special relay	SB	○	—	○	△	×
	Timer	T	—	×	○	○	○
	Retentive timer	ST	—	×	○	○	○
	Counter	C	—	×	○	○	○
	Long counter	LC	—	×	○	○	○
	Data register	D	—	○	○	○	○
	Link register	W	—	○	○	○	○
	Link special register	SW	—	○	○	○	○
System device	Special relay	SM	○	—	○	△	×
	Special register	SD	—	○	○	○	○
Module access device (U□\G□)	Module access device	G	—	○	○	○	○
Index register	Index register	Z	—	×	×	○	×
	Long index register	LZ	—	×	×	○	×
File register	File register	R	—	○	○	○	○
Nesting	Nesting	N	—	—	×	×	×
Pointer	Pointer	P	—	—	○	×	×
	Interrupt pointer	I	—	—	○	×	×
Others	I/O number specified device	U	—	—	○	×	×
Constant	Decimal constant	K	—	—	×	×	×
	Hexadecimal constant	H	—	—	×	×	×
	Real constant	E	—	—	×	×	×
	String constant	—	—	—	×	×	×

Creating comments for module access devices

Device comments can be created within the range of U01 (\G0) to U10 (\G262143).

Appendix 4 Use Project in Different Version

This section explains the considerations for using the projects created with GX Works3 in different versions.
Note each description to use projects.

Compatibility with each module of project

Function	Considerations
Opening projects Writing/reading data to/from a programmable controller Verifying a project/verifying with a programmable controller	These functions cannot be operated in GX Works3 of which version is not supported by the module type.

Using projects with later versions

■Using the projects, which were created in Version 1.010L, in Version 1.015R or later

Function	Considerations
Library operation	Even if the library that contains the global labels created in Version 1.010L is registered into the library list in Version 1.015R or later, the global labels are not displayed in the [Library] tab on the Element Selection window. The global labels called by the utilized elements are changed to the undefined labels without being utilized.

■Using the projects, which were created in Version 1.011M or earlier, in Version 1.015R or later

Function	Considerations
Option setting	For the project created in Version 1.011M or earlier, the following option is set to 'No' if it was opened/read in Version 1.015R or later. For verification, the verification result is inconsistent due to the difference of the option settings. • [Tool] ⇨ [Options] ⇨ "Convert" ⇨ "Basic Setting" ⇨ "Operational Setting" ⇨ "Minimize the Temporary Area".
Verifying with a programmable controller	The verification result may mismatch when verifying the project created in Version 1.011M or earlier with the global label setting in the CPU module. In this case, re-read/re-write from/to the CPU module and then verify with the programmable controller again.

Using projects with earlier versions

■Using the projects, which were created in Version 1.005F or later, in Version 1.002C or earlier

Function	Considerations
Reading from programmable controller	The projects which were created in Version 1.005F or later cannot be read in Version 1.002C or earlier.

■Using the projects, which were created in Version 1.007H or later, in Version 1.006G or earlier

Function	Considerations
Opening projects Reading from a programmable controller	The project that contains the FBD/LD program cannot be opened/read in Version 1.006G or earlier.

■Using the projects, which were created in Version 1.010L or later, in Version 1.008J or earlier

Function	Considerations
Library operation	• FBD/LD elements, which is protected from editing, can be edited by opening the project in Version 1.008J. (For worksheet only) • The edit-protected ST programs will not be the editable elements if the program was opened and copied in Version 1.008J or earlier. To make it editable, open the project that contains the copied data in Version 1.010L or later, and copy it again. • When the edit-protected elements are opened and copied in Version 1.008J or earlier and it ware reopened in Version 1.010L or later, the icons of copied data indicate that some elements are in the state of edit-protections.

■Using the projects, which were created in Version 1.015R or later, in Version 1.010L or earlier

Function	Considerations
Library operation	The read-protected elements, which are utilized from a library, are identified as the elements with a block password if it's opened in Version 1.010L or earlier. Therefore, the deletion of elements/changing names are not possible.

■Using the projects, which were created in Version 1.015R or later, in Version 1.011M or earlier

Function	Considerations
Opening projects Reading from a programmable controller	The project that contains SFC programs cannot be opened/read in Version 1.011M or earlier.
Option setting	The projects that the following options is set to "Yes" cannot be opened/read in Version 1.011M or earlier. In addition, It is not able to verify the projects that the following options is set to "Yes" with the projects created in Version 1.011M or earlier. • [Tool] ⇨ [Options] ⇨ "Convert" ⇨ "Basic Setting" ⇨ "Operational Setting" ⇨ "Minimize the Temporary Area".
Program file setting	When the project set the execution order has been opened and converted in Version 1.011M or earlier, it is converted in a program block name order.
Library operation	Even if the library that contains the global labels is registered into the library list in Version 1.010L, the global labels are not displayed in the [Library] tab on the Element Selection window. The global labels called by the utilized elements are changed to the undefined labels without being utilized.

Appendix 5 Using Simulation Function

Modules supporting the Simulation function are shown below:

• Module type of supported project

Series		Module name
MELSEC iQ-R series	RnCPU	R04CPU
		R08CPU
		R16CPU
		R32CPU
		R120CPU
	RnPCPU	R08PCPU
		R16PCPU
		R32PCPU
		R120PCPU

• Supported modules

Even if the following modules exist in the system configuration of the project, the simulation is performed without being an error.

Series	Module type	
MELSEC iQ-R series	Programmable controller CPU	
	Motion controller CPU	
	Input	
	Output	
	I/O	
	Analog input	
	Analog output	
	Simple motion	
	Pulse I/O, Positioning	
	Information module	Serial communication
		Ethernet
	Network module	CC-Link
		CC-Link IE Field Network
		CC-Link IE Controller Network
Q series	Analog input	
	Analog output	
	Pulse I/O, Positioning	
	Information module	
	Network module	
	Analog I/O	
	Temperature input	
	Temperature control module	
	Loop control	
	Energy measuring module	

Supported CPU module functions

The Simulation function of GX Works3 does not support some CPU module functions.
The following shows the functions that are supported by the Simulation function.

CPU module

Function name					Remarks
Basic function	Program related function	Executing a program (ladder, ST, and FBD/LD)			—
		Scan time monitoring time (WDT) setting			The operation is not performed in actual time.
	Clock function	Clock function	Clock function		The clock data is written to the internal time of a simulator. It differs from the time of a personal computer.
			Time zone setting		The clock of a personal computer is copied to the internal time and used.
		System clock function			The operation is not performed in actual time.
	LED display function	LED display function			—
	Writing to PLC/Reading from PLC	Writing to a programmable controller (including writing data while CPU module is in RUN)			—
		Reading from a programmable controller			
	Diagnostic function	Self-diagnostic function	Error detection function		—
			Module diagnostic function		Host CPU only.
	Monitor and test function	Basic monitor function	Monitoring a program (Circuit monitor)		—
		Test function	Testing device and label data (Changing values)	Testing device and label data (Changing values)	
				Testing local device and local label data	
	Debug function	Changing program and parameter while CPU module is in RUN	Changing program while CPU module is in RUN (Changing ladder block)		—

Function name				Remarks
Operation function	Program related function	Program execution setting function	Program execution type setting function	—
			Device/file usage setting	
		Interrupt function	Fixed cycle interval setting	The operation is not performed in actual time.
			File register block number saving/ recovering setting	—
			Interrupt permission setting during instruction execution	
		PID control function		
		Process control function		—
		Constant scan		The operation is not performed in actual time.
	Operation function	Remote operation function	Operation change function	—
			RUN-PAUSE contact setting function	
	Device/data related function	Device/label memory area setting function (Device variable setting function)	Device/label memory area setting function (Device variable setting function)	—
			Local device setting function	
		Refresh memory setting function		
		Internal buffer capacity setting function		
		Index register setting function		
		Device/label initial value setting function		
		Device comment function		
		File register setting function		
		Timer device limit setting function		
		STOP→RUN operation setting function of output (Y)		
	Communication setting function	Device/label access service processing setting		—
RAS function	Diagnostic function	Self-diagnostic application function	Error clear function	—
	History function	Event history function (Error history/module error history collecting function)		Host CPU only.
Maintenance function	Applied monitor function	(Interrupt) program list monitor		The operation is not performed in actual time.
		Scan time monitor		The operation is not performed in actual time.
		Buffer memory monitor function		—
	Applied test function	Clearing device, label and file register		—
		Clearing latch device and label		

Series-common function

Function name				Remarks
System configuration function	System configuration setting function	Input/output number and point variable setting function		—
Module programming common function	Data communication function	Direct access function (U□\G□, J□\□)		Access with module FB (dedicated instruction) and module label is available.
	Program creation support function	Module label and module FB (device comment) auto-creation function		—
		Sample program display function (Template display function)		
		Parameter auto-creation function (Initial setting and refresh setting)		
RAS function	History function	Event history function	Event history function	—
			(Error history/system error history function)	
	Diagnostic function	Module diagnostics		Host CPU only.

Supported devices

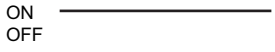
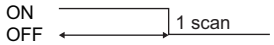
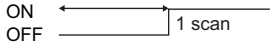
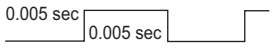
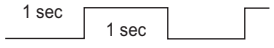
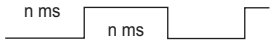
The devices supported by the Simulation function are the same as that of CPU module.

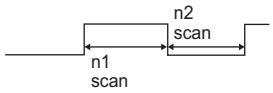
Note that the devices used in an SFC program are not supported.

List of special relay

For details on special relays, refer to the following manual.

MELSEC iQ-R CPU Module User's Manual (Application)

No.	Name	Description
SM0	Latest self diagnostics error (including annunciator ON)	OFF: No error ON: Error state
SM1	Latest self diagnostics error (not including annunciator ON)	OFF: No error ON: Error state
SM50	Error reset	OFF→ON: Error reset request ON→OFF: Error reset complete
SM56	Instruction execution fault	OFF: Normal ON: Instruction execution fault state
SM62	Annunciator	OFF: Not detected ON: Detected
SM80	Detailed information 1: Flag in use	OFF: Not used ON: In use
SM112	Detailed information 2: Flag in use	
SM203	STOP contact	OFF: Other than STOP state ON: STOP state
SM204	PAUSE contact	OFF: Other than PAUSE state ON: PAUSE state
SM210	Clock data set request	OFF→ON: setting request is detected ON→OFF: setting is completed
SM211	Clock data set error	OFF: No error ON: Error state
SM213	Clock data read request	OFF: Non-processing On: Reading request
SM220	CPU No.1 preparation completed	OFF: CPU No.n preparation not completed ON: CPU No.n preparation completed
SM230	CPU No.1 error flag	OFF: CPU No.n normal ON: CPU No.n stop error state
SM400	Always ON	
SM401	Always OFF	
SM402	After RUN, ON for one scan only	
SM403	After RUN, OFF for one scan only	
SM409	0.01 second clock	
SM410	0.1 second clock	
SM411	0.2 second clock	
SM412	1 second clock	
SM413	2 second clock	
SM414	2n second clock	
SM415	2n (ms) clock	

No.	Name	Description
SM420	User timing clock No.0	
SM421	User timing clock No.1	
SM422	User timing clock No.2	
SM423	User timing clock No.3	
SM424	User timing clock No.4	
SM600	Memory card enabled/disabled flag	OFF: Disabled ON: Enabled
SM603	Memory card (drive 2) flag	OFF: No SD memory card inserted ON: SD memory card inserted
SM604	Memory card in-use flag	OFF: Not used ON: In use
SM626	Extended SRAM cassette insertion flag	OFF: Extended SRAM cassette is not attached ON: Extended SRAM cassette is attached
SM628	Program memory write error	OFF: Write error ON: No write operation/normal
SM629	Program memory write flag	OFF: Writing is in progress ON: No write operation
SM632	Data memory write error	OFF: Write error ON: No write operation/normal
SM633	Data memory write flag	OFF: Writing is in progress ON: No write operation
SM699	Dedicated instruction skip flag	OFF: Instruction is executing or completed ON: Instruction has not been executed
SM700	Carry flag	OFF: Carry OFF ON: Carry ON
SM701	Number of output characters selection	OFF: Outputs until reaching NULL code ON: Outputs 16 characters
SM702	Search method	OFF: Sequential search ON: Dichotomizing search
SM703	Sort order	OFF: Ascending ON: Descending
SM704	Block comparison	OFF: Mismatch is detected ON: Completely match
SM709	DT/TM instruction improper data detection flag	OFF: No improper data ON: Improper data is detected
SM752	Dedicated instruction End bit control flag	OFF: End bit automatically controlled ON: End bit not automatically controlled
SM753	File access in progress	OFF: File access is not in progress ON: File access is in progress
SM754	BIN/DBIN instruction error control flag	OFF: Executes error detection ON: No execute error detection
SM755	Scaling data check settings	OFF: Performs data check ON: Not perform data check
SM756	Module access completion wait control flag	OFF: Not wait the completion ON: Waits the completion
SM776	Local device setting at CALL	OFF: Disables local devices ON: Enables local devices
SM777	Local device setting in interrupt programs	OFF: Disables local devices ON: Enables local devices
SM816[*1]	Hold mode	OFF: Value not held ON: Value held
SM817[*1]	Hold mode	OFF: Value not held ON: Value held
SM1200	Auto logging setting file and registration status	OFF: Mismatch ON: Matching
SM1201	SD memory card setting file in use flag	OFF: Not used ON: In use
SM1202	Data memory setting file in use flag	OFF: Not used ON: In use

No.	Name	Description
SM1210	Data logging setting No.1 Data logging preparation	OFF: Not prepared ON: Prepared
SM1211	Data logging setting No.1 Data logging start	OFF: Suspended/waiting for start ON: Start
SM1212	Data logging setting No.1 Data logging data collection in progress	OFF: No data logging collection operation ON: Data logging data collection in progress
SM1213	Data logging setting No.1 Data logging completion	OFF: Data logging is not completed ON: Data logging is completed
SM1214	Data logging setting No.1 Data logging triggering	OFF→ON: triggering occurs
SM1215	Data logging setting No.1 Post data logging triggering	OFF: Not post triggering ON: Post triggering
SM1216	Data logging setting No.1 Data logging error	OFF: No error ON: Error state
SM1217	Data logging setting No.1 Data logging data saving into memory card in progress	OFF: Data saving is not in progress ON: Data saving is in progress
SM1218	Data logging setting No.1 Logging data storage file switching in progress	OFF: Storage file switching is not in progress ON: Storage file switching is in progress
SM1220 to SM1228	Data logging setting No.2	Same configuration as the setting No.1
SM1230 to SM1238	Data logging setting No.3	Same configuration as the setting No.1
SM1240 to SM1248	Data logging setting No.4	Same configuration as the setting No.1
SM1250 to SM1258	Data logging setting No.5	Same configuration as the setting No.1
SM1260 to SM1268	Data logging setting No.6	Same configuration as the setting No.1
SM1270 to SM1278	Data logging setting No.7	Same configuration as the setting No.1
SM1280 to SM1288	Data logging setting No.8	Same configuration as the setting No.1
SM1290 to SM1298	Data logging setting No.9	Same configuration as the setting No.1
SM1300 to SM1308	Data logging setting No.10	Same configuration as the setting No.1
SM1524	Initial processing successful completion state	OFF: The initial processing was completed successfully. ON: The initial processing was not completed successfully.
SM1525	Initial processing error completion state	OFF: The initial processing was completed with an error. ON: The initial processing was not completed with an error.

*1 RnPCPU only supports it.

List of special register

For details on special registers, refer to the following manual.

MELSEC iQ-R CPU Module User's Manual (Application)

No.	Name	Description
SD0	Latest self diagnostics error code	Latest self diagnostics error code
SD1	Clock time for self diagnosis error occurrence	Clock time for self diagnosis error occurrence
SD2		
SD3		
SD4		
SD5		
SD6		
SD7		
SD10	Self diagnostics error code	Self diagnostics error code 1
SD11		Self diagnostics error code 2
SD12		Self diagnostics error code 3
SD13		Self diagnostics error code 4
SD14		Self diagnostics error code 5
SD15		Self diagnostics error code 6
SD16		Self diagnostics error code 7
SD17		Self diagnostics error code 8
SD18		Self diagnostics error code 9
SD19		Self diagnostics error code 10
SD20		Self diagnostics error code 11
SD21		Self diagnostics error code 12
SD22		Self diagnostics error code 13
SD23		Self diagnostics error code 14
SD24		Self diagnostics error code 15
SD25		Self diagnostics error code 16
SD62	Annunciator number	Annunciator number
SD63	Number of annunciators	Number of annunciators
SD64 to SD79	Table of detected annunciator numbers	Detected annunciator number
SD80	Detailed information 1 information category	Detailed information 1 information category code
SD81 to SD111	Detailed information 1	Detailed information 1
SD112	Detailed information 2 information category	Detailed information 2 information category code
SD113 to SD143	Detailed information 2	Detailed information 2
SD200	Switch status	CPU switch status
SD201	LED status	CPU-LED state
SD203	CPU module operating status	CPU module operating status
SD210	Clock data	Clock data (year)
SD211		Clock data (month)
SD212		Clock data (day)
SD213		Clock data (hour)
SD214		Clock data (minute)
SD215		Clock data (second)
SD216		Clock data (day of the week)
SD218	Time zone setting value	Time zone (in minutes)
SD241	Number of extension base units	0: base unit only 1 to 7: number of extension base units
SD242	Identification for whether or not Q series module can be mounted	Identification of the base type 0: Q series module cannot be mounted (There is no base unit that can mount the Q series unit) 1: Q series module can be mounted (There is a base unit that can mount the Q series unit)

No.	Name	Description
SD243	Number of base slots	Number of base slots
SD244		
SD250	Latest I/O for implemented unit	Latest I/O number for implemented unit
SD260	Number of points assigned to bit devices	X number of points assigned (L)
SD261		X number of points assigned (H)
SD262		Y number of points assigned (L)
SD263		Y number of points assigned (H)
SD264		M number of points assigned (L)
SD265		M number of points assigned (H)
SD266		B number of points assigned (L)
SD267		B number of points assigned (H)
SD268		SB number of points assigned (L)
SD269		SB number of points assigned (H)
SD270		F number of points assigned (L)
SD271		F number of points assigned (H)
SD272		V number of points assigned (L)
SD273		V number of points assigned (H)
SD274		L number of points assigned (L)
SD275		L number of points assigned (H)
SD280	Number of points assigned to word devices	D number of points assigned (L)
SD281		D number of points assigned (H)
SD282		W number of points assigned (L)
SD283		W number of points assigned (H)
SD284		SW number of points assigned (L)
SD285		SW number of points assigned (H)
SD288	Number of points assigned to timer-type devices	T number of points assigned (L)
SD289		T number of points assigned (H)
SD290		ST number of points assigned (L)
SD291		ST number of points assigned (H)
SD292		C number of points assigned (L)
SD293		C number of points assigned (H)
SD294		LT number of points assigned (L)
SD295		LT number of points assigned (H)
SD296		LST number of points assigned (L)
SD297		LST number of points assigned (H)
SD298		LC number of points assigned (L)
SD299		LC number of points assigned (H)
SD300	Number of points assigned to index registers	Z number of points assigned
SD302	Number of points assigned to long index registers	LZ number of points assigned
SD306	Number of points assigned to file registers	ZR number of points assigned (L)
SD307		ZR number of points assigned (H)
SD308	Number of points assigned to refresh devices	RD number of points assigned (L)
SD309		RD number of points assigned (H)
SD312	File register block number	File register block number
SD412	One second counter	Number of counts in 1-second units
SD414	2n second clock setting	Unit setting for 2n second clock
SD415	2n ms clock setting	Unit setting for 2n ms clock
SD420	Scan counter	Number of counts in each scan
SD500	Execution program number	Execution type of the program being executed
SD518	Initial scan time	Initial scan time (unit: ms)
SD519		Initial scan time (unit: μs)
SD520	Current scan time	Current scan time (unit: ms)
SD521		Current scan time (unit: μs)

No.	Name	Description
SD522	Minimum scan time	Minimum scan time (unit: ms)
SD523		Minimum scan time (unit: μs)
SD524	Maximum scan time	Maximum scan time (unit: ms)
SD525		Maximum scan time (unit: (μs)
SD526	END processing time	END processing time (unit: ms)
SD527		END processing time (unit: μs)
SD528	Constant scan waiting time	Constant scan waiting time (unit: ms)
SD529		Constant scan waiting time (unit: μs)
SD530	Scan program execution time	Scan program execution time (unit: ms)
SD531		Scan program execution time (unit: μs)
SD600	Memory card mounting status	SD memory card type
SD604	SD memory card (drive 2) usage status	SD memory card (drive 2) usage status
SD606	SD memory card (drive 2) capacity	SD memory card (drive 2) capacity: the lower digit (unit: KB)
SD607		SD memory card (drive 2) capacity: the higher digits (unit: KB)
SD610	SD memory card (drive 2) free space	SD memory card (drive 2) free space: the lower digit (unit: KB)
SD611		SD memory card (drive 2) free space: the higher digits (unit: KB)
SD614	Device/label memory (drive 3) usage status	Device/label memory (drive 3) usage status
SD616	Device/label memory (drive 3) capacity	Device/label memory (drive 3) capacity: the lower digits (unit: KB)
SD617		Device/label memory (drive 3) capacity: the higher digits (unit: KB)
SD618	Device/label memory (file storage area) capacity	Device/label memory (file storage area) (drive 3) capacity: the lower digits (unit: KB)
SD619		Device/label memory (file storage area) (drive 3) capacity: the lower digits (unit: KB)
SD620	Data memory (drive 4) usage status	Data memory (drive 4) usage status
SD622	Data memory (drive 4) capacity	Data memory (drive 4) capacity: the lower digits (unit: KB)
SD623		Data memory (drive 4) capacity: the higher digits (unit: KB)
SD626	Extended SRAM cassette capacity identification information	Capacity identification information of the Extended SRAM cassette
SD629	Program memory write (transfer) status	Write (transfer) status display (percent)
SD633	Data memory write (transfer) status	Write (transfer) status display (percent)
SD757	Current interrupt priority	Current interrupt priority
SD758	Interrupt disabling for each priority setting value	Interrupt disabling for each priority setting value
SD771	Specification of the number of write instruction executions to data memory	Specification of the number of write instruction executions to data memory
SD816*1	Execution cycle	Execution cycle
SD817*1		
SD818*1	Bumpless function availability setting for the S.PIDP instruction	0: Enable 1: Disable
SD819*1	Dummy device	Dummy device
SD820*1		
SD1210	Data logging setting No.1 Latest storage file number	Latest storage file number
SD1211		
SD1212	Data logging setting No.1 Oldest storage file number	Oldest storage file number
SD1213		
SD1214	Data logging setting No.1 Internal buffer free space	Internal buffer free space size (KB)
SD1215	Data logging setting No.1 Number of processing overflow occurrences	Number of processing overflow occurrences
SD1216	Data logging setting No.1 Data logging error cause	Data logging error cause
SD1220 to SD1226	Data logging setting No.2	Same configuration as the setting No.1
SD1230 to SD1236	Data logging setting No.3	Same configuration as the setting No.1
SD1240 to SD1246	Data logging setting No.4	Same configuration as the setting No.1
SD1250 to SD1256	Data logging setting No.5	Same configuration as the setting No.1
SD1260 to SD1266	Data logging setting No.6	Same configuration as the setting No.1

No.	Name	Description
SD1270 to SD1276	Data logging setting No.7	Same configuration as the setting No.1
SD1280 to SD1286	Data logging setting No.8	Same configuration as the setting No.1
SD1290 to SD1296	Data logging setting No.9	Same configuration as the setting No.1
SD1300 to SD1306	Data logging setting No.10	Same configuration as the setting No.1
SD1400 to SD1463	Interrupt pointer mask pattern	Mask pattern
SD1504	Open completion signal	Open completion
SD1505	Open request signal	Open request

*1 RnPCPU only supports it.

Supported instructions

The instructions supported by the Simulation function are shown below:

Note that no processing is performed for unsupported instructions (NOP processing).

Sequence instruction

Category	Symbol
Contact instruction	AND, ANDF, ANDFI, ANDP, ANDPI, ANI, LD, LDF, LDFI, LDI, LDP, LDPI, OR, ORF, ORFI, ORI, ORP, ORPI
Association instruction	ANB, EGF, EGP, INV, MEF, MEP, MPP, MPS, MRD, ORB
Output instruction	FF, OUT, OUT C, OUT F, OUT LC, OUT LT/LST, OUT T/ST, OUTH T/ST, PLF, PLS, RST, RST F, SET, SET F
Shift instruction	SFT(P)
Master control instruction	MC, MCR
Termination instruction	END, FEND
Stop instruction	STOP
No operation instruction	NOP, NOPLF

Basic instruction

Category	Symbol*1
Comparison operation instruction	AND□(_U), ANDD□(_U), BKCMP□(P)(_U), DBKCMP□(P)(_U), LD□(_U), LDD□(_U), OR□(_U), ORD□(_U)
Arithmetic operation instruction	*(P)(_U), +(P)(_U), -(P)(_U), /(P)(_U), B*(P), B+(P), B-(P), B/(P), BK+(P)(_U), BK-(P)(_U), D*(P)(_U), D+(P)(_U), D-(P)(_U), D/(P)(_U), DB*(P), DB+(P), DB-(P), DB/(P), DBK+(P)(_U), DBK-(P)(_U), DDEC(P)(_U), DEC(P)(_U), DINC(P)(_U), INC(P)(_U)
Data transfer instruction	BLKMOVB(P), BMOV(P), BMOVL(P), BXCH(P), CML(P), CMLB(P), DCML(P), DFMOV(P), DFMOVL(P), DMOV(P), DXCH(P), FMOV(P), FMOVL(P), MOV(P), MOVB(P), SWAP(P), XCH(P)
Logical operation instruction	BKAND(P), BKOR(P), BKXNR(P), BKXOR(P), DAND(P), DOR(P), DXNR(P), DXOR(P), WAND(P), WOR(P), WXNR(P), WXOR(P)
Data shift instruction	BSFL(P), BSFR(P), DSFL(P), DSFR(P), SFL(P), SFR(P), SFTBL(P), SFTBR(P), SFTWL(P), SFTWR(P)
Bit processing instruction	BKRST(P), BRST(P), BSET(P), DTEST(P), TEST(P)
Data conversion instruction	ASC2INT(P), BCD(P), BIN(P), BKBCD(P), BKBIN(P), BTOW(P), DABCD(P), DABIN(P)(_U), DBCD(P), DBIN(P), DBL2DINT(P), DBL2INT(P), DBL2UDINT(P), DBL2UINT(P), DDABCD(P), DDABIN(P)(_U), DECO(P), DGBIN(P)(_U), DGRY(P)(_U), DHABIN(P), DINT2INT(P), DINT2UDINT(P), DINT2UINT(P), DIS(P), DNEG(P), DVAL(P)(_U), EMOD(P), ENCO(P), FLT2DINT(P), FLT2INT(P), FLT2UDINT(P), FLT2UINT(P), GBIN(P)(_U), GRY(P)(_U), HABIN(P), INT2DINT(P), INT2UDINT(P), INT2UINT(P), NDIS(P), NEG(P), NUNI(P), SEG(P), UDINT2DINT(P), UDINT2INT(P), UDINT2UINT(P), UINT2DINT(P), UINT2INT(P), UINT2UDINT(P), UNI(P), VAL(P)(_U), WTOB(P)

*1 □ indicates an operator.

Application instruction

Category	Symbol*1
Program branch instruction	CJ, GOEND, JMP, SCJ
Program execution control instruction	DI, EI, IMASK, IRET, SIMASK, WDT(P)
Rotation instruction	DRCL(P), DRCR(P), DROL(P), DROR(P), RCL(P), RCR(P), ROL(P), ROR(P)
Data processing instruction	DMAX(P)(_U), DMEAN(P)(_U), DMIN(P)(_U), DSERDATA(P), DSORTD(_U), DSUM(P), DWSUM(P)(_U), MAX(P)(_U), MEAN(P)(_U), MIN(P)(_U), SERDATA(P), SORTD(_U), SUM(P), WSUM(P)(_U)
Structure creation instruction	BREAK(P), CALL(P), ECALL(P), EFCALL(P), FCALL(P), FOR, NEXT, RET, XCALL
Data table operation instruction	FDEL(P), FIFR(P), FIFW(P), FINS(P), FPOP(P)

Category	Symbol*1
Debugging and failure diagnostic instruction	LEDR
String processing instruction	$+(P), $MOV(P), $MOV(P)_WS, AND$□, BCDDA(P), BINDA(P)(_U), BINHA(P), DBCDDA(P), DBINDA(P)(_U), DBINHA(P), DSTR(P)(_U), ESTR(P), INSTR(P), INT2ASC(P), LD$□, LEFT(P), LEN(P), MIDR(P), MIDW(P), OR$□, RIGHT(P), SJIS2WS(P), SJIS2WSB(P), STR(P)(_U), STRDEL(P), STRINS(P), WS2SJIS(P)
Data control instruction	BAND(P)(_U), DBAND(P)(_U), DLIMIT(P)(_U), DSCL(P)(_U), DSCL2(P)(_U), DZONE(P)(_U), LIMIT(P)(_U), SCL(P)(_U), SCL2(P)(_U), ZONE(P)(_U)
File register operation instruction	ADRSET(P), QDRSET(P), RSET(P), ZRRDB(P), ZRWRB(P)
Clock instruction	ANDDT□, ANDTM□, DATE+(P), DATE-(P), DATERD(P), DATEWR(P), LDDT□, LDTM□, ORDT□, ORTM□, S(P).DATE+, S(P).DATE-, S(P).DATERD, SEC2TIME(P), TIME2SEC(P)
Program control instruction	POFF(P), PSCAN(P), PSTOP(P)
PID control instruction	PIDCONT(P), PIDINIT(P), PIDPRMW(P), PIDRUN(P), PIDSTOP(P), S(P).PIDCONT, S(P).PIDINIT, S(P).PIDPRMW, S(P).PIDRUN, S(P).PIDSTOP
Reading/writing data instruction	S(P).DEVLD, SP.DEVST, SP.FREAD, SP.FWRITE
Real number instruction	ACOS(P), ACOSD(P), ANDE□, ANDED□, ASIN(P), ASIND(P), ATAN(P), ATAND(P), BACOS(P), BASIN(P), BATAN(P), BCOS(P), BDSQRT(P), BSIN(P), BSQRT(P), BTAN(P), COS(P), COSD(P), DBL2FLT(P), DEG(P), DEGD(P), DINT2DBL(P), DINT2FLT(P), E*(P), E+(P), E-(P), E/(P), ED*(P), ED+(P), ED-(P), ED/(P), EDMAX(P), EDMIN(P), EDMOV(P), EDNEG(P), EDSQRT(P), EMAX(P), EMIN(P), EMOV(P), ENEG(P), EREXP(P), ESQRT(P), EVAL(P), EXP(P), EXPD(P), FLT2DBL(P), INT2DBL(P), INT2FLT(P), LDE□, LDED□, LOG(P), LOG10(P), LOG10D(P), LOGD(P), ORE□, ORED□, POW(P), POWD(P), RAD(P), RADD(P), SIN(P), SIND(P), TAN(P), TAND(P), UDINT2DBL(P), UDINT2FLT(P), UINT2DBL(P), UINT2FLT(P)
Random number instruction	RND(P), SRND(P)
Index register instruction	ZPOP(P), ZPUSH(P)
Special counter instruction	UDCNT1, UDCNT2
Special timer instruction	STMR, TTMR
Shortcut control instruction	ROTC
Ramp signal instruction	RAMPQ
Pulse related instruction	PLSY, PWM, SPD
Matrix input instruction	MTR
Direct read/write in 1-byte unit	ZRRDB(P), ZRWRB(P)
Indirect address read	ADRSET(P)
Timing check instruction	DUTY, TIMCHK
Module access instruction	DFROM(P), DFROMD(P), DTO(P), DTOD(P), FROM(P), FROMD(P), TO(P), TOD(P), TYPER(D), UNIINFRD(P)
Logging instruction	LOGTRG, LOGTRGR
Process control instruction (RnPCPU only)	S.2PID, S.□, S.ABS, S.ADD, S.AMR, S.AT1, S.AVE, S.BC, S.BPI, S.BUMP, S.D, S.DBND, S.DED, S.DIV, S.DUTY, S.ENG, S.FG, S.FLT, S.HS, S.I, S.IENG, S.IFG, S.IN, S.IPD, S.LIMT, S.LLAG, S.LS, S.MID, S.MOUT, S.MUL, S.ONF2, S.ONF3, S.OUT1, S.OUT2, S.PGS, S.PHPL, S.PID, S.PIDP, S.PSUM, S.R, S.SEL, S.SPI, S.SQR, S.SUB, S.SUM, S.TPC, S.VLMT1, S.VLMT2

*1 □ indicates an operator.

Enabled/disabled parameter items

The Simulation function does not support some parameter setting items.
The following shows the parameter setting items that are supported by the Simulation function.

System parameter

Setting item		
I/O Assignment Setting	Base/Power/Extension Cable Setting	Slots
	I/O Assignment Setting	Module Name
		Module Status Setting
		Module/Points/Start XY
		Control PLC Setting
	Setting of Points Occupied by Empty Slot	
Multiple CPU Setting	Number of CPU modules	
	Communication Setting between CPU	Fixed Scan Communication Function
		Fixed Scan Communication Area Setting

CPU parameter

Setting item			
Operation Related Setting	Timer Limit Setting		
	RUN-PAUSE Contact Setting		
	Remote Reset Setting		
	Output Mode Setting of STOP → RUN		
	Clock Related Setting		
Interrupt Settings	Fixed Scan Interval Setting		
	Fixed Scan Execution Mode Setting		
	Interrupt Enable Setting in Executing Instruction		
	Block No. Save/Recovery Setting		
Service Processing Setting	Device/Label Access Service Processing Setting		
File Setting	File Register Setting		
	Initial Value Setting		
	File Setting for Device Data Storage		
Memory/Device Setting	Device/Label Memory Area Setting	Extended SRAM Cassette Setting	
		Device/Label Memory Area Capacity Setting	
		Device/Label Memory Area Detailed Setting	Points
			Local Device
	Index Register Setting		
	Refresh Memory Setting		
	Pointer Setting		
	Internal Buffer Capacity Setting		
RAS Setting	Scan Time Monitoring Time (WDT) Setting		
	Constant Scan Setting		
	Error Detections Setting		
	CPU Module Operation Setting at Error Detected		
	LED Display Setting		
	Event History Setting		
Program Setting	Program Setting	Program Setting	Program Name
			Execution Type
			Detailed Setting Information (Fixed Scan)
			Detailed Setting Information (Event)
			Device/File Use or not
	FB/FUN File Setting		

Module buffer memory

The following shows the number of points of buffer memory (U□\G□) for a module supported by the Simulation function.

Series	Module type		Points
MELSEC iQ-R series	Programmable controller CPU		524288
	Motion controller CPU		524288
	Input		512
	Output		512
	I/O		512
	Analog input		131072
	Analog output		131072
	Simple motion		131072
	Pulse I/O, Positioning		131072
	Information module	Serial communication	65536
		Ethernet	2097152
	Network module	CC-Link	32768
		CC-Link IE Field Network	65536
		CC-Link IE Controller Network	2097152
Q series	Analog input		131072
	Analog output		
	Pulse I/O and Positioning		
	Information module		
	Network module		
	Analog I/O		
	Temperature input		
	Temperature control module		
	Loop control		
	Energy measuring module		

Link devices of network module

The following shows the link devices and the number of points for network module supported by the Simulation function.

Series	Module type		Device type	Points
MELSEC iQ-R series	Network module	Ethernet (When CC-Link IE Field Network is used)	J□\SB	512
			J□\SW	512
			J□\X	16384
			J□\Y	16384
			J□\W	16384
		CC-Link IE Controller Network	J□\SB	512
			J□\SW	512
			J□\X	8192
			J□\Y	8192
			J□\B	32768
			J□\W	131072
		CC-Link IE Field Network	J□\SB	512
			J□\SW	512
			J□\X	16384
			J□\Y	16384
			J□\W	16384

Considerations

The following describes the considerations on debugging programs using the Simulation function.

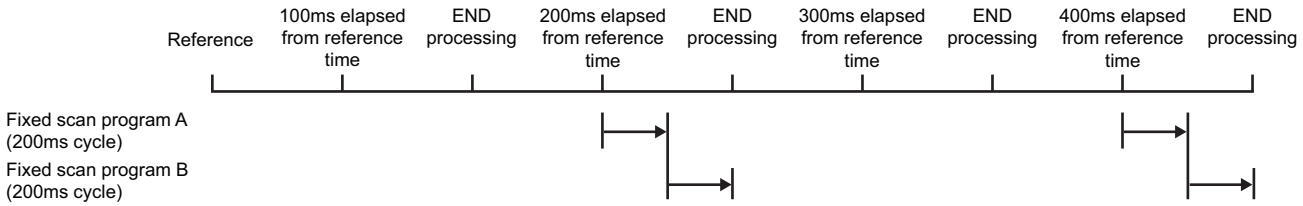
Simulation time

Since the instruction processing speed differs between the Simulation function and CPU module, the simulator operates as follows.

- The simulation time is calculated by adding the average execution time per one instruction, which is calculated based on the PC MIX value, at every internal instruction execution of a CPU module.
  The simulator operates behind the actual time when the simulation time takes longer than the actual time due to the performance of the personal computer. The simulation time gains in accordance with the actual time when the actual time takes longer than the simulation time.

Operation of fixed cycle execution program

The program is executed in accordance with the simulation time.
The timer and the fixed cycle interrupt are the same.



Handling of floating point

A rounding error may occur in the operation result of an instruction using floating point. The result will not the same as that of a CPU module.

Module

■I/O module

Simulation of the operation does not support I/O module.

■Intelligent function module

The buffer memory for Intelligent function module is reserved by the system parameter setting, "I/O Assignment Setting".
The buffer memory (U☐\G☐) cannot be accessed when the I/O Assignment Setting is not configured.
By selecting an Intelligent function module as the module type and setting the number of points in the I/O Assignment Setting, the buffer memory according to the specified number of points is created.
If the module type is not specified, it becomes an empty slot.
If the module type is specified but the number of point is not specified, the specified number of points for the module becomes equivalent to one specified for "Setting of Points Occupied by Empty Slot" of system parameter.

Network

The Simulation function does not support the network.

Watchdog timer

It operates in accordance with the simulation time.

Attachment of extended SRAM cassette

The simulator operates as if an extended SRAM cassette (8 MB) was attached.
Parameter error will occur when specifying the capacity more than 8 MB on the "Extended SRAM Cassette Setting" of CPU parameter.

Drive usage

The data in the ROM drive of CPU module (program memory/data memory) is stored in a temporary folder in the hard disk of a personal computer.

The ROM drive usage per data depends on the hard disk of a personal computer.

Enabling the remote RESET

The simulator operates as if "Enable" was set to "Remote Reset" in the Simulation function regardless of "Remote Reset Setting" under "Operation Related Setting" of the CPU parameter.

Writing parameters to Intelligent function module

When writing parameters on the "Online Data Operation" screen, the availability differs according to the interaction with an Intelligent function module.

○: Writable, ×: Not writable, —: Not supported

Parameter	Cooperated	Not cooperated
Simple motion module setting	○	×
Other than above	—	×

CPU number

The Simulation function only supports CPU No.1.

The Simulation function is performed by handling CPU No.1 as the host CPU when the host CPU is set to CPU No.2 to CPU No.4 in "I/O Assignment Setting" of the system parameter.

Ex.

The host CPU is set to CPU No.4 in "I/O Assignment Setting".

CPU number	I/O Assignment Setting	Simulation
CPU No.1	R04CPU	R32CPU (host CPU)
CPU No.2	R08CPU	R08CPU
CPU No.3	R16CPU	R16CPU
CPU No.4	R32CPU (host CPU)	R32CPU

Clock function

It operates in accordance with the simulation time.

The clock of a personal computer is reflected to the time of the clock function every time the CPU module is reset.

Appendix 6 USB Driver Installation Procedure

To communicate with a CPU module via USB, installing a USB driver is required.
The procedure of USB driver installation is as shown below.
If multiple MELSOFT products are installed previously, refer to their installed location.

Windows® XP

Operating procedure

1. Connect the personal computer and a CPU module with a USB cable, and power ON the programmable controller.
2. Select "Install from a list or specific location (Advanced)" on the "Found New Hardware Wizard" screen.
3. Select "Search for the best driver in these locations" on the next screen. Select "Include this location in the search", and specify 'MELSOFT\Easysocket\USBDrivers' in the folder where GX Works3 has been installed.

Precautions

If the driver cannot be installed, check the following settings on Windows®.
The USB driver may not be installed when "Block—Never install unsigned driver software" is selected in [Control Panel] ⇨ [System] ⇨ [Hardware] ⇨ [Driver Signing].
Select "Ignore — Install the software anyway and don't ask for my approval", or "Warn — Prompt me each time to choose an action" in [Driver Signing], and execute the USB driver installation.

Windows Vista®

Operating procedure

1. Connect the personal computer and a CPU module with a USB cable, and power ON the programmable controller.
2. Select "Locate and install driver software (recommended)" on the "Found New Hardware Wizard" screen.
3. Select "Browse my computer for driver software (advanced)" on the "Found New Hardware" screen.
4. Select "Search for the best driver in these locations" on the displayed screen. Select "Include subfolders" and specify 'MELSOFT\Easysocket\USBDrivers' in the folder where GX Works3 has been installed.

Precautions

When "Windows can't verify the publisher of this driver software" is displayed during the installation, select "Install this driver software anyway".

Windows® 7 or later

Operating procedure

1. Connect the personal computer and a CPU module with a USB cable, and power ON the programmable controller.
2. Select [Control Panel] ⇨ [System and Security] ⇨ [Administrative Tools] ⇨ [Computer Management] ⇨ [Device Manager] from Windows® Start*1. Right-click "Unknown device" and click "Update Driver Software".
3. Select "Browse my computer for driver software" on the "Update Driver Software" screen, and specify 'MELSOFT\Easysocket\USBDrivers' in the folder where GX Works3 has been installed on the next screen.

*1 Select [All apps] on the Start screen or [Start] ⇨ [All Programs].

MEMO

INDEX

B

Block password 293

C

Calculating memory size 190
Clearing to zero 331
Clock 325
Color 48
Comment 135
Common device comment 176
Current value 277

D

Data 58
Data Logging 323
Device intial value 197
Device memory 191

E

Each program device comment 176
Event history 322
Execution count of interrupt program 281

F

File password 307
Font 48
Function 133,148,217
Function block 128,148,214

G

Global device 120

I

I/O assignment setting 95
Initialization 331
Inline structured text 134

L

Latch clear 331
Local device 120

M

MC 135
Memory dump 284
Module FB 216
Module label 115
Monitoring 266
Multiple CPU setting 95

N

Network configuration 97
NOP 139
Note 138

O

Offline monitor 288

P

Processing time of program 280
Profile 83

R

Refresh device 98
Remote operation 326
Remote password 310
RUN/STOP switch 326

S

Sample comment 182
Security key authentication 295,300
Shortcut key 47
Single file format 57
Statement 136
Synchronous setting 95
System monitor 313

T

Title 74

U

USB driver 357
User authentication 303
User data 262

W

Watch 277
Workspace format 56

MEMO

REVISIONS

*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
June 2014	SH(NA)-081215ENG-A	First edition
July 2014	SH(NA)-081215ENG-B	Partial correction
October 2014	SH(NA)-081215ENG-C	■Added or modified parts Section 1.2, Section 2.4, Section 2.8, Section 2.9, Section 3.1, Section 3.2, Section 4.2, Section 4.3, Section 4.4, Section 5.1, Section 5.2, Section 5.4, Section 6.1, Section 6.5, Section 12.3, Chapter 13, Section 13.2, Section 13.4, Section 13.8, Section 15.2, Section 15.3, Section 15.4, Section 16.1, Section 17.3, Appendix 1
January 2015	SH(NA)-081215ENG-D	■Added or modified parts TERMS, Chapter 1, Section 1.1, Section 1.2, Section 1.3, Section 1.4, Section 2.3, Section 2.4, Section 2.9, Section 3.1, Section 3.2, Section 3.3, Section 3.4, Section 3.5, Chapter 4, Section 4.1, Section 4.2, Section 4.3, Section 4.4, Section 5.1, Section 5.2, Section 5.3, Section 5.4, Section 6.1, Section 6.2, Section 6.3, Section 6.4, Section 6.5, Section 6.6, Section 7.1, Section 8.1, Chapter 9, Section 9.1, Section 9.2, Section 9.3, Section 9.4, Section 9.5, Section 9.6, Section 9.7, Chapter 10, Section 10.1, Section 10.2, Chapter 11, Section 11.1, Section 11.2, Section 12.1, Section 12.2, Section 12.3, Section 12.4, Section 12.5, Section 12.6, Section 12.7, Section 13.1, Section 13.3, Section 13.4, Chapter 14, Section 14.1, Section 14.2, Section 14.5, Section 14.6, Section 14.7, Section 14.8, Section 15.3, Section 15.4, Section 15.5, Chapter 16, Section 16.1, Section 16.2, Section 16.3, Section 16.4, Section 18.1, Section 18.2, Section 18.4, Appendix 1, Appendix 3, Appendix 4
April 2015	SH(NA)-081215ENG-E	■Added or modified parts TERMS, Section 2.3, Section 2.4, Section 2.9, Section 3.2, Section 3.3, Section 3.4, Section 3.5, Section 3.6, Section 4.2, Section 4.3, Section 4.4, Section 6.1, Section 6.2, Section 6.3, Section 6.7, Chapter 10, Section 10.1, Section 10.3, Section 10.4, Section 13.2, Section 15.4, Chapter 16, Section 16.2, Section 16.3, Appendix 1, Appendix 4
July 2015	SH(NA)-081215ENG-F	■Added or modified parts TERMS, Section 1.2, Section 1.3, Section 1.4, Section 2.1, Section 2.3, Section 2.4, Section 2.7, Section 2.9, Section 3.1, Section 3.2, Section 3.3, Section 3.4, Section 3.5, Section 3.6, Section 3.7, Section 4.1, Section 4.2, Section 4.3, Section 5.1, Section 5.2, Section 5.4, Section 6.1, Section 6.2, Section 6.3, Section 6.5, Section 6.6, Section 6.8, Section 6.9, Section 6.10, Chapter 9, Section 9.1, Section 9.2, Section 9.3, Section 9.4, Section 9.5, Section 9.6, Section 9.7, Chapter 10, Section 10.1, Section 10.3, Section 10.4, Chapter 11, Section 11.3, Section 12.3, Section 13.1, Section 13.3, Section 13.6, Chapter 14, Section 14.1, Section 14.2, Section 14.3, Section 14.4, Section 14.9, Chapter 15, Section 15.3, Section 15.4, Section 15.5, Section 15.6, Section 15.7, Section 16.2, Section 16.4, Chapter 18, Section 18.3, Section 18.5, Appendix 1, Appendix 2, Appendix 3, Appendix 4, Appendix 5

Japanese manual number: SH-081214-F

This manual confers no industrial property rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2014 MITSUBISHI ELECTRIC CORPORATION

TRADEMARKS

Microsoft, Windows, Windows Vista, Windows NT, Windows XP, Windows Server, Visio, Excel, PowerPoint, Visual Basic, Visual C++, and Access are either registered trademarks or trademarks of Microsoft Corporation in the United States, Japan, and other countries.
Intel, Pentium, and Celeron are either registered trademarks or trademarks of Intel Corporation in the United States and other countries.
Ethernet is a trademark of Xerox Corp.
The SD and SDHC logos are either registered trademarks or trademarks of SD-3C, LLC.
MODBUS is a registered trademark of Schneider Electric SA.
All other company names and product names used in this manual are either trademarks or registered trademarks of their respective companies.

SH(NA)-081215ENG-F(1507)
MODEL: GXW3-O-E

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.